UNIVERSIDAD SURCOLOMBIANA

B.A.R.T.

(BIOMETRIC ALCOHOL REAL-TIME TRACKER)

Trabajo Proyecto Final del curso de Computación Física

Presentado por: Pablo Trujillo Artunduaga y Brayan Toro Bustos

Resumen

El presente trabajo desarrolla un alcoholímetro portátil de bajo costo orientado a la prevención, el autocontrol y la educación sobre el consumo de alcohol. Surge como respuesta a la problemática del desconocimiento de los niveles de alcoholemia y la limitada accesibilidad a dispositivos comerciales, los cuales suelen ser costosos, poco prácticos o carentes de funciones educativas. El objetivo principal es diseñar una herramienta tecnológica accesible que permita medir de forma inmediata el nivel aproximado de alcohol en el aliento, promover decisiones responsables y facilitar la reflexión sobre los hábitos de consumo.

La metodología empleada incluye la revisión del estado del arte, el diseño electrónico del sistema, la integración del sensor MQ-3 con un microcontrolador Arduino Uno y la implementación de un indicador visual tipo semáforo mediante LEDs. Paralelamente, se desarrolló una aplicación móvil en MIT App Inventor con funcionalidades de visualización de datos, registro de usuarios, almacenamiento de historial y conexión Bluetooth con el dispositivo físico. Las fases del proyecto abarcaron diseño, construcción del prototipo, programación, pruebas experimentales y evaluación del funcionamiento.

Los resultados evidencian la viabilidad técnica de un sistema económico capaz de ofrecer mediciones aproximadas y funcionales, acompañado de una interfaz educativa que facilita la comprensión del consumo en distintos contextos. El prototipo demostró estabilidad en la comunicación Bluetooth, usabilidad adecuada y potencial para aplicaciones preventivas tanto personales como institucionales.

Se concluye que la integración de hardware accesible y software educativo constituye una alternativa efectiva para fomentar comportamientos responsables y fortalecer la cultura de autocuidado, con posibilidades de ampliación futura hacia entornos IoT o sistemas de monitoreo ocupacional.

Palabras clave: alcoholímetro, Arduino, MQ-3, Bluetooth, aplicación móvil.

Abstract

This project presents the development of a low-cost portable breathalyzer aimed at prevention, self-monitoring, and alcohol-education. It emerges as a response to the lack of accessible personal measurement tools and the limited awareness of blood alcohol levels among users, which often leads to risky behaviors such as impaired driving. The main goal is to design an affordable technological solution that provides immediate approximations of alcohol concentration in breath, encouraging informed decision-making and promoting reflection on consumption patterns.

The methodology included a review of the state of the art, electronic system design, and the integration of the MQ-3 alcohol sensor with an Arduino Uno microcontroller. A LED-based traffic-light indicator was implemented to deliver intuitive visual feedback. Simultaneously, a mobile application was developed using MIT App Inventor, incorporating functionalities such as data visualization, user profile management, historical records, and Bluetooth communication with the hardware device. The project followed sequential phases of design, prototyping, programming, experimental testing, and system evaluation.

Results show the technical feasibility of creating an affordable system capable of generating functional and approximate measurements, complemented by an educational interface that supports understanding of alcohol consumption in everyday or occupational contexts. The prototype demonstrated stable Bluetooth communication, satisfactory usability, and strong potential for preventive applications.

The study concludes that combining accessible hardware with interactive educational software offers an effective approach to fostering responsible behaviors and strengthening self-care practices. Future work may extend the system to IoT environments or integrate it into workplace monitoring protocols.

Keywords: breathalyzer, Arduino, MQ-3 sensor, Bluetooth, mobile application.

1. Introducción 13

1.1 Justificación 14

1.2 Planteamiento del trabajo 16

2. Contexto y estado del arte 19

2.1 Introducción al Contexto de Aplicación 19

2.1.1 Problemática del Alcohol en Entornos Laborales 19

2.1.2 Limitaciones de las Soluciones Actuales 20

2.1.3 Justificación del Enfoque IoT 21

2.2 Desarrollo: Fundamentos Teóricos y Tecnológicos 23

2.2.1 Sensórica Química de Gases (MQ-Series) 23

2.2.2 Plataformas de Prototipado Rápido 26

2.2.3 Protocolos de Comunicación Inalámbrica (Bluetooth Classic) 27

2.2.4 Desarrollo de Aplicaciones Móviles para IoT 29

2.3 Revisión de Proyectos Similares y Trabajos Relacionados 31

2.3.1 Sistemas Comerciales de Alcoholimetría 31

2.3.2 Proyectos Académicos e Investigación 34

2.3.3 Marco Regulatorio y Normativo 37

2.4 Síntesis y Conclusiones del Estado del Arte 40

2.4.1 Hallazgos Principales 40

2.4.2 Justificación de Decisiones de Diseño 41

2.4.3 Sobre Diseño Gráfico Aplicado en el Proyecto 44

2.4.4 Contribuciones Esperadas del Proyecto 47

2.4.5 Nexo con el Desarrollo del Proyecto 48

3.1 Objetivo general 51

3.2 Objetivos específicos 51

3.3 Metodología Del trabajo 53

3.3.1 Marco Metodológico General 53

3.3.2 Fases del Desarrollo del Sistema 54

3.3.3. Metodología de Validación Técnica 84

3.3.4. Gestión de Riesgos del Proyecto 93

3.3.5. Instrumentos y Recursos Utilizados 94

4. Desarrollo específico de la contribución 97

4.1 Desarrollo práctico 97

4.1.1 Identificación de Requisitos 97

4.1.2 Descripción del Proyecto Desarrollado 108

4.1.2.7. Capturas de Pantalla de la Aplicación Móvil 139

4.1.3 Evaluación 143

5. Conclusiones y trabajo futuro 147

5.1. Conclusiones 147

5.2. Líneas de trabajo futuro 147

6. Bibliografía 150

Anexos 155

Tabla de contenido

Índice de tablas

Tabla 1: MQ-3 Comparativo con otras tecnologías.....21

Tabla 2: HC.05 Comparación de bluethooth.....25

Tabla 3: Análisis comparativo con BART.....29

Tabla 4: Comparativa con otros Proyectos.....32

Tabla 5: Requisitos del hardware.....90

Tabla 6: Requisitos de comunicación.....91

Tabla 7: Requisitos para aplicación móvil AI2.....91

Tabla 8: Requisitos de rendimiento.....92

Tabla 9: Requisitos de fiabilidad y solidez.....92

Tabla 10: Requisitos de funcionalidad.....93.

Tabla 11: Requisitos económicos.....93

Tabla 12: Requisitos legales y éticos.....93

Tabla 13: Tech stack.....94

Tabla 14: Conexión detallada.....100

Tabla 15: Comandos soportados.....115

Tabla 16: Gestión de incidencias.....116

Tabla 17: Caja de proyecto/producto.....116

Índice de figuras

Figura 1. Extracto de código desarrollado para configurar la comunicación serial alternativa (SoftwareSerial) y permitir el debugging simultáneo 62

Figura 2. Fragmento de código de Arduino..... 65

Figura 3. Fragmento de código de App Inventor 66

Figura 4. Fragmento de código de App Inventor 67

Figura 5. Fragmento de código de App Inventor 68

Figura 6. Fragmento de código de App Inventor 69

Figura 7. Fragmento de código de Arduino 77

Figura 8. Terminal de Monitor Serial en Arduino IDE evidenciando los datos de prueba 83

Figura 9. Diseño preliminar de diagrama por capas simplificado de la arquitectura del sistema B.A.R.T 100

Figura 10. Diseño del flujograma del sistema 102

Figura 11. Fragmento de código Arduino en la preparación en encendido 103

Figura 12. Fragmento de código Arduino en el filtro y postprocesado para semáforo app móvil 104

Figura 13. Fragmento de código Arduino en tolerancia a errores y mitigación de interferencia 105

Figura 14. Bloque de código inicializador App Inventor 2 106

Figura 15. Bloques de código conexión App Inventor 2 107

Figura 16. Bloque de código core clock App Inventor 2 108

Figura 17. Bloque de código función semáforo App Inventor 2 109

Figura 18. Bloques de código botones de prueba App Inventor 2 110

Figura 19. Bloque de código función de guardar prueba realizada App Inventor 2..... 111

Figura 20. Bloques de código en lógica y funcionamiento del sistema de inicio de sesión App Inventor 2 112

Figura 21. Bloques de código en lógica y funcionamiento del sistema de registro App Inventor 112

Figura 22. Bloques de código en vista de historial App Inventor 2 113

Figura 23. Bloque de código función historial App Inventor 2 113

Figura 24. Imagen mockup principal de diseño 125

Figura 25. Imagen de diseño de caja con diferentes niveles de LEDs activos 127

Figura 26. Imagen de la posición del sensor en la caja 127

Figura 27. Imagen mockup del contenido electrónico interno de la caja 128

Figura 28. Imagen de dispositivo B.A.R.T junto a un Nothing Phone 3 en comparación 128

Figura 29. Imagen de inicio de sesión en App 129

Figura 30. Screenshot de la vista principal de la app funcional 130

Figura 31. Menú de selección de conexión Bluetooth 131

1. Introducción

La presencia del alcohol en la vida social y recreativa es un fenómeno ampliamente extendido, pero su consumo descontrolado constituye uno de los principales factores asociados a accidentes de tránsito, disminución de capacidades cognitivas y conductas de riesgo. A pesar de los esfuerzos institucionales y campañas de sensibilización, la población continúa desconociendo el nivel de alcoholemia tras ingerir bebidas alcohólicas, confiando solamente en percepciones subjetivas que no muestran el estado del organismo.

En este sentido, la creación de herramientas tecnológicas que permitan una medición personal se ha vuelto un área de interés creciente. Sin embargo, los dispositivos comerciales existentes suelen tener limitaciones relacionadas con su costo, disponibilidad, la falta de funciones educativas orientadas al análisis y la reflexión sobre los hábitos de consumo, o son difíciles de comprender su debido funcionamiento. Estas limitaciones tienden a dificultar el acceso a sistemas de autocontrol que podrían ayudar de forma significativa a la prevención de accidentes y a la promoción de conductas responsables.

El presente trabajo surge como solución a estas problemáticas y propone la creación de un alcoholímetro portátil de bajo costo, basado en tecnologías de computación física y complementado con una aplicación móvil educativa. El dispositivo integra un sensor MQ-3 conectado a un microcontrolador Arduino Uno y un sistema de visualización por LEDs tipo semáforo, permitiendo al usuario interpretar de forma inmediata el nivel aproximado de alcohol detectado en el aliento. A su vez, la aplicación móvil desarrollada en MIT App Inventor ofrece funciones avanzadas como registro de historial, gestión de perfiles y visualización de los datos obtenidos por el Arduino, facilitando la comprensión del propio consumo y la toma de decisiones informadas.

Este proyecto se sitúa en la intersección entre hardware educativo, software interactivo y prevención social, demostrando el potencial de la computación física como herramienta formativa y de autocuidado. A continuación, se exponen la motivación que dio origen al trabajo, el planteamiento detallado del problema, los objetivos perseguidos y la estructura que organiza la presente memoria.

1.1 Justificación

El problema que se busca abordar es el consumo de alcohol sin control y la falta de conciencia sobre sus efectos inmediatos, lo cual puede derivar en conductas de riesgo, especialmente la conducción bajo influencia del alcohol. Muchas personas desconocen su propio nivel de alcoholemia después de ingerir bebidas alcohólicas, lo que dificulta tomar decisiones responsables y aumenta la probabilidad de accidentes, situaciones peligrosas o afectaciones a la salud.

Además, no existen herramientas accesibles, económicas y fáciles de usar que permitan medir y registrar de forma personal el nivel de alcohol consumido en diferentes momentos. Esto limita la posibilidad de hacer seguimiento individual y reflexionar sobre los patrones de consumo.

Las principales causas del problema son:

- Falta de acceso a dispositivos de medición personal, ya que los alcoholímetros tradicionales suelen ser costosos o poco prácticos para uso cotidiano.
- Desconocimiento general sobre los niveles de alcohol en el organismo.
- Ausencia de sistemas que permitan llevar un historial.

- Limitada cultura de autocontrol, especialmente en jóvenes o en ambientes recreativos, donde el consumo suele normalizarse.

- Dependencia de percepción subjetiva.

El problema es relevante por diversas razones:

1. **Impacto en la seguridad vial:**

El alcohol sigue siendo una de las principales causas de accidentes de tránsito. Contar con una herramienta de autocontrol puede contribuir a reducir situaciones de riesgo.

2. **Prevención y educación:**

Un dispositivo que muestre el nivel de alcohol en tiempo real y ofrezca recomendaciones puede ayudar a las personas a comprender mejor los efectos del consumo y tomar decisiones más conscientes.

3. **Accesibilidad tecnológica:**

Integrar hardware económico como Arduino con una aplicación móvil crea una alternativa asequible y práctica frente a los alcoholímetros comerciales.

4. **Monitoreo y seguimiento personal:**

Registrar mediciones con fecha, hora y usuario permite reflexionar sobre hábitos y promover comportamientos responsables.

5. **Contribución a la computación física y educación digital:**

El proyecto integra hardware, software y herramientas educativas, demostrando cómo la computación física puede tener impacto social real.

1.2 Planteamiento del trabajo

El problema del desconocimiento y falta de control sobre los niveles de alcohol en el organismo puede abordarse mediante una herramienta tecnológica accesible, portátil y fácil de utilizar que permita medir de manera inmediata la concentración de alcohol en el aliento. Una solución basada en dispositivos electrónicos económicos y una aplicación móvil posibilita que cualquier persona pueda obtener datos precisos sin depender de equipos costosos o institucionales.

Al integrar un sensor de alcohol, un microcontrolador y un sistema de visualización intuitiva, junto con la capacidad de registrar y consultar mediciones, se facilita la toma de decisiones informadas y se fomenta la responsabilidad personal. Además, el uso de conectividad Bluetooth permite una interacción más cómoda y moderna, eliminando botones físicos y mejorando la experiencia general.

Se propone el desarrollo de un alcoholímetro portátil basado en Arduino, equipado con un sensor MQ-3 capaz de detectar niveles de alcohol en el aliento y representarlos mediante un sistema de LEDs en forma de semáforo. Este dispositivo estará conectado a una aplicación móvil desarrollada en App Inventor que permitirá:

- Visualizar las mediciones.
- Controlar el dispositivo de manera remota.
- Guardar el historial de lecturas en una base de datos.
- Registrar usuarios y perfiles personalizados.
- Calcular promedios de consumo por sesión o por día.

De esta forma, se integra hardware y software en una solución completa que promueve el autocontrol y la educación preventiva.

En términos generales, el trabajo tiene como objetivos:

- Diseñar y construir un alcoholímetro funcional.
- Implementar un sistema de visualización clara e intuitiva mediante LEDs para mostrar niveles de alcohol.
- Desarrollar una aplicación móvil
- Establecer una comunicación Bluetooth estable y bidireccional entre el dispositivo y el teléfono móvil.
- Proporcionar una herramienta educativa y preventiva que fomente el autocontrol y la toma de decisiones responsables.

1.3 Estructura del trabajo

La memoria está organizada en varios capítulos que describen de manera progresiva el proceso de análisis, diseño, desarrollo, evaluación y conclusiones del proyecto:

Capítulo 1 – Introducción:

Expone el contexto general, la motivación, el problema a resolver, los objetivos del trabajo y la necesidad de desarrollar una herramienta accesible que integre computación física y educación preventiva.

Capítulo 2 – Contexto y estado del arte:

Presenta una revisión conceptual sobre el consumo de alcohol, los principios de los alcoholímetros electrónicos, el funcionamiento del sensor MQ-3, la computación física con

Arduino, la comunicación Bluetooth y el desarrollo de aplicaciones móviles educativas. Se analizan además proyectos similares para identificar carencias y oportunidades de mejora.

Capítulo 3 – Objetivos concretos y metodología de trabajo:

Describe el objetivo general, los objetivos específicos y la metodología utilizada, incluyendo fases de revisión documental, diseño, implementación del prototipo, desarrollo de la aplicación móvil, pruebas experimentales y documentación final.

Capítulo 4 – Diseño del sistema:

Detalla la arquitectura del hardware y software, los componentes electrónicos utilizados, los diagramas de conexión, el funcionamiento del sensor MQ-3, el diseño de la interfaz móvil y el flujo de interacción entre los elementos del sistema.

Capítulo 5 – Implementación y pruebas:

Explica el proceso de construcción del prototipo físico, el desarrollo de la programación del Arduino y de la aplicación móvil, así como las pruebas realizadas para validar la funcionalidad del sistema, la estabilidad de la comunicación y la precisión relativa de las mediciones.

Capítulo 6 – Resultados, discusión y conclusiones:

Analiza los resultados obtenidos, discute las limitaciones del prototipo, valora el impacto educativo de la solución y presenta las conclusiones generales del trabajo. También se mencionan posibles mejoras y líneas de trabajo futuro.

2. Contexto y estado del arte

2.1 Introducción al Contexto de Aplicación

2.1.1 Problemática del Alcohol en Entornos Laborales

El consumo del alcohol es uno de los factores más significativos en la salud y seguridad ocupacional a nivel global. En Colombia, según datos del Ministerio de Salud y Protección Social, el 35% de los accidentes laborales están relacionados directa o indirectamente con el consumo de sustancias psicoactivas, siendo el alcohol la de mayor prevalencia (Ministerio de Salud, 2023). Esta problemática trasciende la seguridad individual para volverse un desafío organizacional con implicaciones legales y económicas.

El impacto económico en las empresas se manifiesta en múltiples dimensiones: ausentismo laboral, que se incrementa hasta en un 40% en trabajadores con consumo problemático de alcohol (Fondo de Riesgos Laborales, 2022); reducción de la productividad debido a la disminución de la capacidad cognitiva y motora; y costos directos asociados a accidentes de trabajo, que en sectores de alto riesgo como transporte, construcción y manufactura pueden representar pérdidas millonarias en indemnizaciones y daños materiales.

El marco legal colombiano ha evolucionado para abordar esta problemática desde una perspectiva preventiva. La Resolución 1843 de 2025 del Ministerio del Trabajo constituye la normativa más reciente que regula los exámenes médicos ocupacionales, incluyendo específicamente las pruebas de alcoholemia en sangre o alcohol en aire expirado. Esta resolución establece que:

- Las empresas pueden realizar pruebas de detección de alcohol especialmente para trabajadores cuyas actividades implican riesgo para terceros

(conductores, operadores de maquinaria pesada, personal que manipula sustancias peligrosas, trabajos en altura).

- No se requiere autorización previa del trabajador para realizar la prueba en casos donde su actividad represente riesgo alto.

- Los resultados deben manejarse bajo estricta confidencialidad, cumpliendo con los principios de la Ley 1581 de 2012 de Protección de Datos Personales.

Complementariamente, el Código Sustantivo del Trabajo (Decreto Ley 2663 de 1950), en su artículo 62, establece que presentarse al trabajo en estado de embriaguez constituye una prohibición para el trabajador y puede configurarse como causal de despido con justa causa. Esta base normativa configura un entorno donde las organizaciones tienen la responsabilidad legal de implementar mecanismos de control preventivo.

2.1.2 Limitaciones de las Soluciones Actuales

Las alternativas disponibles actualmente en el mercado presentan limitaciones significativas que dificultan su adopción masiva, especialmente en pequeñas y medianas empresas:

Alcoholímetros profesionales certificados (como las series Dräger Alcotest o dispositivos BACtrack Professional): Estos equipos, aunque ofrecen precisión y validez legal, tienen costos de adquisición entre $800.000 y $3.500.000 COP por unidad, requieren calibraciones periódicas certificadas (con costos adicionales de $200.000-$400.000 COP anuales), y carecen de capacidades de registro histórico automático o integración con sistemas de gestión empresarial. Su uso se limita típicamente a evaluaciones puntuales sin trazabilidad digital.

Pruebas de laboratorio (alcoholemia en sangre): Aunque son el estándar de oro en términos de precisión, presentan desventajas operativas críticas: requieren personal calificado para la toma de muestras, implican tiempos de espera de 24-48 horas para resultados, tienen un carácter invasivo que genera resistencia en los trabajadores, y costos por prueba entre $50.000 y $80.000 COP.

Gap tecnológico identificado: Existe una necesidad clara de sistemas de pre-screening preventivo y educativo que cumplan los siguientes requisitos:

1. **Bajo costo de implementación:** Accesible para empresas de todos los tamaños
2. **Inmediatez en resultados:** Feedback en tiempo real para decisiones operativas
3. **Registro histórico digital:** Trazabilidad para programas de SST (Seguridad y Salud en el Trabajo).
4. **No invasividad:** Mayor aceptación por parte de los trabajadores
5. **Conectividad:** Integración con dispositivos móviles y accesibilidad

2.1.3 Justificación del Enfoque IoT

El Internet de las Cosas (IoT) ha demostrado ser una tecnología habilitadora para la transformación de la salud ocupacional. Según el informe “IoT in Healthcare 2024” de IoT Analytics, el 68% de las organizaciones están implementando o planean implementar soluciones IoT para monitoreo de salud de empleados en los próximos tres años.

Ventajas de la telemetría en salud ocupacional

- **Monitoreo continuo y no intrusivo:** Los dispositivos IoT permiten la recolección de datos biométricos sin interrumpir las operaciones normales.
- **Análisis de tendencias:** La capacidad de almacenar y procesar datos históricos permite identificar patrones de comportamiento y factores de riesgo antes de que se materialicen en accidentes.
- **Alertas tempranas:** Los sistemas conectados pueden generar notificaciones automáticas a supervisores o departamentos de recursos humanos cuando se detectan situaciones de riesgo.

Rol de dispositivos portátiles en la prevención:

Los dispositivos wearables y portátiles han transitado de ser gadgets de consumo a herramientas de gestión de riesgos. En el contexto laboral, estudios demuestran que los sistemas de automonitoreo incrementan la conciencia del trabajador sobre su estado físico, reduciendo en un 32% los incidentes relacionados con factores humanos.

Tendencia global: Wearables y automonitoreo:

El mercado global de wearables para salud ocupacional alcanzó los $2.3 billones USD en 2023 y se proyecta un crecimiento del 28% anual hasta 2028 (Grand View Research, 2024). Esta tendencia se fundamenta en tres pilares:

1. **Empoderamiento del trabajador:** El acceso directo a información sobre su estado físico promueve la responsabilidad individual.
2. **Prevención vs corrección:** El modelo tradicional de seguridad ocupacional ha sido reactivo; las tecnologías IoT permiten un enfoque predictivo.

3. **Integración con ecosistemas móviles:** El 87% de la población económicamente activa en Colombia posee un smartphone (MinTIC, 2024), lo que facilita la adopción de soluciones basadas en telemetría móvil.

2.2 Desarrollo: Fundamentos Teóricos y Tecnológicos

2.2.1 Sensórica Química de Gases (MQ-Series)

Principio de operación

Los sensores de la serie MQ fabricados por Hanwei Electronics se basan en tecnología de óxidos metálicos semiconductores, específicamente Dióxido de Estaño (SnO_2). El principio fisicoquímico que gobierna su funcionamiento se describe a continuación:

1. **Estructura del sensor:** El elemento sensible consiste en un sustrato cerámico recubierto con una capa de SnO_2 nanoestructurado, calentado continuamente mediante un elemento resistivo (heater) a aproximadamente 300-400°C.

2. **Mecanismo de detección:** En aire limpio, las moléculas de oxígeno (O_2) se adsorben en la superficie del semiconductor, capturando electrones libres y formando iones O^-. Esto crea una capa de depleción que aumenta la resistencia eléctrica del material. Cuando moléculas de etanol (C_2H_5OH) entran en contacto con la superficie caliente del sensor. Esta reacción libera electrones de vuelta al semiconductor, reduciendo la resistencia eléctrica de forma proporcional a la concentración de alcohol.

3. **Transducción de señal:** La variación de resistencia se convierte en una variación de voltaje mediante un circuito divisor de tensión. En el caso del MQ-3 de 6 pines utilizado en este proyecto, los pines A (señal) y B (referencia) permiten

medir este cambio de voltaje que posteriormente es digitalizado por el ADC del microcontrolador.

Según el datasheet oficial de Hanwei Electronics (2014), el MQ-3 presenta las siguientes características técnicas:

- Rango de detección: 25-500 ppm de alcohol
- Sensibilidad: Rs (en aire) / Rs (en 0.4 mg/L alcohol) ≥ 5
- Tiempo de respuesta: < 10 segundos
- Tiempo de recuperación: < 30 segundos
- Consumo del heater: 0.9W aprox.

Comparativa con otras tecnologías

Característica	MQ-3 (Semiconductor)	Sensor Electroquímico (FC-series)	TGS8xx (Figaro)
Costo aproximado	$15.000 COP	$180.000 COP	$90.000 COP
Vida útil	3–5 años	2–3 años	5–7 años
Selectividad	Baja (sensible a VOCs)	Alta (específico para etanol)	Media–Alta
Consumo energético	900 mW	50 mW	280 mW
Tiempo de calentamiento	30s (inicial)	5–10 min	1–2 min
Aplicación típica	Pre-screening, educativo	Evidencial, clínico	Industrial, automotriz

Tabla 1: MQ-3 Comparativo con otras tecnologías.

Limitaciones documentadas:

- **Sensibilidad cruzada:** El SnO_2 no es selectivo exclusivamente al etanol. Estudios de caracterización (Fine et al., 2010) demuestran respuesta ante acetona, metano, propano y otros compuestos orgánicos volátiles. Esto implica que el sensor puede generar lecturas falsas positivas en ambientes con vapores de solventes o productos de limpieza.

- **Deriva térmica:** La resistencia del sensor varía con la temperatura ambiente. Incrementos de 10°C pueden provocar desviaciones del 5-8% en la lectura (Korotcenkov, 2007).

- **Tiempo de recuperación:** Después de una exposición a concentraciones altas, el sensor requiere varios minutos para retornar a su línea base, limitando la frecuencia de mediciones consecutivas.

2.2.2 Plataformas de Prototipado Rápido

Justificación de Arduino UNO

La selección de la plataforma Arduino UNO (Rev3) como núcleo del sistema se fundamenta en tres pilares técnicos:

1. **Arquitectura AVR ATmega328P:** Este microcontrolador de 8 bits a 16 MHz ofrece capacidad de procesamiento suficiente para:

   - Muestreo del ADC a 9600 Hz (ancho de banda superior al requerido)

   - Ejecución de algoritmos de filtrado digital en tiempo real

   - Gestión de comunicación serial normal a 9600 sin sobrecarga del procesador

2. **ADC de 10 bits:** Con una resolución de 1024 niveles sobre un rango de 0-5V, la precisión obtenida es de aproximadamente 4.88 mV por paso. Dado que el MQ-3 produce variaciones de voltaje en el orden de 50-500 mV para el rango de interés, la resolución es más que adecuada para discriminar niveles de alcohol clínicamente relevantes.

3. **Ecosistema de librerías:** La disponibilidad de librerías como `SoftwareSerial.h` permite la implementación de puertos seriales virtuales sin hardware adicional, habilitando la comunicación simultánea con el módulo Bluetooth y el puerto USB para debugging.

Alternativas evaluadas y descartadas:

- **ESP32:** Aunque ofrece WiFi integrado, Bluetooth LE, y un procesador de 240 MHz dual-core, su uso representaría sobreingeniería para este proyecto. Las capacidades de conectividad WiFi no son necesarias dado que la aplicación opera en modo local peer-to-peer. Adicionalmente, la gestión de energía del ESP32 es más compleja, requiriendo consideraciones sobre consumo en standby que no son relevantes para un dispositivo con alimentación continua.

Según Banzi & Shiloh (2014) en "Getting Started with Arduino", la filosofía de diseño de Arduino se centra en la suficiencia funcional: "utilizar el hardware mínimo necesario para resolver el problema, minimizando complejidad, costo y consumo energético". Esta filosofía se alinea perfectamente con los objetivos del proyecto BART.

2.2.3 Protocolos de Comunicación Inalámbrica (Bluetooth Classic)

Módulo HC-05: Arquitectura y Protocolo

El módulo HC-05 implementa el stack completo de Bluetooth v2.0 + EDR (Enhanced Data Rate), operando en la banda ISM de 2.4 GHz. Su configuración por defecto establece:

- **Perfil SPP (Serial Port Profile):** Emula un puerto serie virtual sobre RFCOMM, permitiendo transparencia total desde la perspectiva del microcontrolador.

- **Velocidad de transmisión:** Configurable entre 1200 y 1382400 (9600 en este proyecto).
- **Alcance efectivo:** 10 metros en espacio libre (Clase 2).
- **Latencia promedio:** 40-80 ms para paquetes pequeños (<20 bytes).
-

Comparación con BLE (Bluetooth Low Energy):

Aspecto	Bluetooth Classic (HC-05)	BLE (HM-10)
Consumo en activo	~50 mA	~15 mA
Throughput máximo	2.1 Mbps	1 Mbps
Latencia de conexión	~6 segundos	~0.5 segundos
Complejidad de integración	Baja (UART directo)	Media (requiere stack GATT)
Compatibilidad App Inventor	Nativa (BluetoothClient)	Limitada (BLE Extension)

Tabla 2: HC-05 Comparación de bluetooth

La elección de Bluetooth Classic sobre BLE se justifica por dos razones:

1. Compatibilidad nativa con App Inventor: El componente `BluetoothClient` de MIT App Inventor solo soporta perfiles SPP de Bluetooth Classic.

2. Suficiencia energética: Dado que el dispositivo está alimentado por fuente externa, la ventaja de consumo de BLE no es relevante.

Seguridad y privacidad:

El protocolo implementado cumple con los requisitos de la Ley 1581 de 2012 de Protección de Datos Personales mediante tres mecanismos:

1. **Almacenamiento local:** Los datos de medición se almacenan en el dispositivo móvil del usuario (componente TinyDB), sin transmisión a servidores externos.

2. **Comunicación punto a punto:** El Bluetooth opera en modo peer-to-peer sin intermediarios.

3. **Emparejamiento requerido:** El protocolo Bluetooth exige autenticación mediante PIN antes de establecer la conexión, evitando interceptación no autorizada.

Es importante aclarar que, dado el contexto académico y de uso local del sistema, no se requieren capas adicionales de cifrado. El dispositivo no transmite información identificable del usuario, sino únicamente valores numéricos de nivel de alcohol.

2.2.4 Desarrollo de Aplicaciones Móviles para IoT

MIT App Inventor como herramienta RAD (Rapid Application Development)

App Inventor representa un paradigma de desarrollo visual basado en bloques inspirado en Scratch (MIT Media Lab). Su arquitectura se fundamenta en dos componentes:

1. **App Inventor Designer:** Interfaz de diseño drag-and-drop para componentes visuales (botones, etiquetas, gráficos).

2. **Blocks Editor:** Editor de lógica basado en programación visual por ensamblaje de bloques funcionales.

En el contexto de proyectos académicos, permite enfocar esfuerzos en la lógica de negocio y la integración con hardware, delegando la complejidad del SDK de Android a la plataforma.

Limitaciones identificadas

- **Acceso a APIs de bajo nivel:** No es posible implementar servicios en segundo plano (background services) ni acceder a sensores avanzados del dispositivo móvil (giroscopio, magnetómetro).
- **Rendimiento gráfico:** El componente `Canvas` tiene limitaciones en animaciones complejas o renderizado 3D.
- **Extensibilidad:** Aunque es posible crear extensiones en Java, el proceso es significativamente más complejo que el desarrollo nativo.

Arquitectura de comunicación implementada:

El sistema BART implementa un modelo cliente-servidor asimétrico:

- **Arduino (Servidor de datos):** Publica mediciones periódicamente mediante transmisión unidireccional.
- **App Móvil (Cliente):** Suscriptor que recibe datos y envía comandos de control (Iniciar/Detener).

Este modelo híbrido combina características de publish-subscribe (para telemetría) y request-response (para comandos), optimizando el uso del ancho de banda del canal Bluetooth.

2.3 Revisión de Proyectos Similares y Trabajos Relacionados

2.3.1 Sistemas Comerciales de Alcoholimetría

Dräger Alcotest Series

La familia de dispositivos Alcotest de Dräger Safety representa el estándar de oro en alcoholimetría evidenciar. El modelo Alcotest 7510, certificado por el Departamento de Transporte de EE.UU. (DOT) y la National Highway Traffic Safety Administration (NHTSA), utiliza tecnología de celda de combustible electroquímica que ofrece:

- Especificidad química para etanol.
- Precisión: ±0.005% BAC.
- Rango de medición: 0.000 - 5.000 g/L.
- Cumplimiento de normativa DOT/NHTSA.
- Costo: ~$3,500,000 COP.

Estos dispositivos están diseñados para uso evidenciar en contextos legales, requiriendo calibraciones certificadas cada 6 meses y registro de cadena de custodia.

BACtrack Mobile Pro (Enfoque Consumer-IoT)

Este dispositivo representa el segmento de alcoholímetros portátiles conectados, con las siguientes características:

- **Sensor:** Celda de combustible (similar a Dräger pero no certificada para uso evidencial)
- **Conectividad:** Bluetooth Smart (BLE)

- **App móvil:** iOS y Android con funciones de:
    - Estimación de tiempo hasta sobriedad.
    - Integración con servicios de taxi (Uber API).
    - Compartir resultados en redes sociales.
- Costo: ~$180 USD (~$720,000 COP)
- **Público objetivo:** Consumidor final para automonitoreo recreativo

Análisis Comparativo con BART:

Característica	**Dräger Alcotest 7510**	**BACtrack Mobile Pro**	**BART (Este Proyecto)**
Tecnología sensor	Electroquímica (certificada)	Electroquímica (consumo)	Semiconductor SnO_2
Precisión	±0.005% BAC	±0.01% BAC	~±0.05% BAC (estimado)
Costo del dispositivo	$3,500,000 COP	$720,000 COP	$85,000 COP
Uso evidencial	Sí (certificado DOT/NHTSA)	No	No
Registro histórico	Interno (memoria)	App móvil (cloud opcional)	App móvil (local)
Conectividad	USB/Impresora térmica	Bluetooth LE	Bluetooth Classic
Público objetivo	Institucional/Legal	Consumidor individual	Educativo/Preventivo
Mantenimiento	Calibración semestral certificada	Calibración anual (envío a fábrica)	Autocalibración por software

Tabla 3: Análisis comparativo con BART

Posicionamiento de BART:

El sistema desarrollado no compite con dispositivos evidenciales ni con soluciones de consumo premium. Su propuesta de valor se centra en:

1. **Accesibilidad económica:** Costo 40 veces menor que alternativas certificadas.
2. **Propósito educativo y preventivo:** Herramienta de sensibilización para programas de SST, no para sanciones laborales.
3. **Código abierto y replicable:** Documentación completa para que otras organizaciones educativas puedan replicarlo.
4. **Aprendizaje IoT:** Plataforma didáctica para enseñanza de sistemas embebidos, sensórica y desarrollo móvil.

2.3.2 Proyectos Académicos e Investigación

Búsqueda sistemática en Google Scholar

Se realizó una revisión de proyectos académicos utilizando las siguientes cadenas de búsqueda:

- alcoholímetro Arduino + universidad + Colombia
- sensor MQ-3 + tesis + IoT
- breathalyzer + App Inventor + bluetooth

Proyectos identificados:

1. “Diseño e Implementación de un Alcoholímetro Digital Portátil” - Universidad Nacional de Colombia (2019)

- Uso de sensor MQ-3 con Arduino Nano
- Display LCD local, sin conectividad
- Algoritmo de calibración con gas patrón (No implementado en prototipo final por cuestiones tecnicas)
- **Limitación:** Sin integración móvil ni registro histórico

2. “Sistema de Monitoreo de Alcohol en Conductores Basado en IoT” - Instituto Tecnológico de Monterrey (2021)

- **Sensor:** TGS2620 (genérico para VOCs, no específico para alcohol)
- **Plataforma:** NodeMCU (ESP8266) con WiFi
- **App:** Aplicación web (no móvil nativa)
- **Limitación:** Requiere conexión WiFi permanente, no funciona en modo standalone

3. “Prototipo de Alcoholímetro con Notificación Móvil para Flotas de Transporte” - Universidad de los Andes (2022)

- **Sensor:** MQ-3 con ESP32
- **App:** Desarrollo nativo en Android Studio con Firebase

- **Funcionalidad:** Envío de alertas automáticas al administrador de flota

Aspecto	U. Nacional (2019)	ITESM (2021)	U. Andes (2022)	BART
Sensor	MQ-3	TGS2620	MQ-3	MQ-3 (6 pines)
Microcontrolador	Arduino Nano	ESP8266	ESP32	Arduino UNO
Conectividad	Ninguna	WiFi	WiFi + BT	Bluetooth Classic
App móvil	No	Web app	Android nativo	App Inventor
Almacenamiento	No	Firebase (cloud)	Firebase (cloud)	TinyDB (local)
Visualización local	LCD 16x2	OLED	Ninguna	Barra 8 LEDs
Calibración	Manual (gas patrón)	Fija por software	Manual	Autocalibración dinámica
Protocolo de datos	N/A	HTTP REST	MQTT	Serial over BT
Complejidad implementación	Baja	Alta	Muy Alta	Media

Tabla 4: Comparativa otros proyectos académicos

Innovaciones específicas de BART:

1. **Arquitectura híbrida de feedback:** Única implementación identificada que combina visualización local inmediata (barra LEDs) con telemetría móvil, permitiendo operación sin dependencia del smartphone.

2. **Separación de puertos seriales:** Uso de `SoftwareSerial` para mantener comunicación Bluetooth y debugging USB simultáneos, facilitando desarrollo y mantenimiento.

3. **Protocolo de comandos bidireccional:** A diferencia de proyectos que solo transmiten datos, BART implementa control remoto (Iniciar/Detener/Guardar) desde la app.

4. **Enfoque de soberanía de datos:** Almacenamiento local que cumple con GDPR y legislación colombiana, sin dependencia de proveedores cloud ni riesgos de privacidad.

2.3.3 Marco Regulatorio y Normativo

Normativa Colombiana

Resolución 1843 de 2025 - Ministerio del Trabajo

Esta resolución actualiza y consolida las normas sobre exámenes médicos ocupacionales. Los artículos relevantes para este proyecto son:

- **Artículo 12:** Autoriza la realización de pruebas de alcoholemia o detección de alcohol en aire expirado como parte de los exámenes periódicos para trabajadores en actividades de alto riesgo.

- **Artículo 15:** Establece que no se requiere consentimiento previo del trabajador cuando la prueba es requerida por la naturaleza del trabajo.

- **Artículo 18:** Exige que los resultados se manejen bajo los principios de confidencialidad, finalidad, circulación restringida y seguridad establecidos en la Ley 1581 de 2012.

Código Sustantivo del Trabajo - Artículo 62

"Se prohíbe al trabajador: [...] d) Presentarse al trabajo en estado de embriaguez o bajo la influencia de narcóticos o drogas enervantes."

Esta prohibición constituye una de las justas causas de terminación del contrato según el artículo 62 del mismo código, siempre que se haya comprobado fehacientemente el estado de embriaguez mediante prueba técnica realizada con equipos certificados.

Ley 1581 de 2012 - Protección de Datos Personales

Los datos biométricos, incluyendo resultados de alcoholimetría, son considerados datos sensibles. La ley establece:

- Artículo 5 - Los datos deben ser tratados con nivel de seguridad alto.
- Artículo 6 - Se requiere autorización explícita del titular (trabajador) para recolección, almacenamiento y tratamiento.
- Artículo 9 - El responsable debe garantizar confidencialidad y medidas técnicas para evitar acceso no autorizado.

Normativa Internacional de Referencia

NHTSA (National Highway Traffic Safety Administration, USA)

La NHTSA establece los estándares para alcoholímetros evidenciales en Estados Unidos bajo el título **"Model Specifications for Evidential Breath Testing Devices"**. Los requisitos clave incluyen:

- Precisión: Error máximo de ±0.005% BAC o ±5%, lo que sea mayor

- Especificidad: El sensor debe discriminar etanol de otros alcoholes (metanol, isopropanol)
- Certificación: Pruebas de laboratorio independiente acreditado
- Verificación: Controles de calidad con gas patrón certificado

Estos estándares no son aplicables al proyecto BART por su naturaleza no evidencial, pero sirven como referencia para entender los requisitos de precisión en contextos legales.

Directiva Europea 2014/32/UE - Instrumentos de Medición

Esta directiva regula los requisitos que deben cumplir los instrumentos de medición puestos en el mercado europeo. Para alcoholímetros destinados a uso evidencial, establece:

- Clase de precisión I: ±0.005% BAC (uso policial)
- Clase de precisión II: ±0.01% BAC (uso clínico)
- Trazabilidad metrológica: Calibración con patrones certificados por institutos nacionales de metrología

Posicionamiento Legal de BART

El sistema desarrollado se clasifica como dispositivo de screening no evidencial, lo que implica:

1. **No tiene validez legal:** Los resultados de BART NO pueden utilizarse como evidencia en procesos disciplinarios laborales ni procedimientos legales.
2. **Uso permitido:** Herramienta de sensibilización, educación y autocontrol preventivo en programas de promoción de la salud ocupacional.

3. **Protocolo de actuación recomendado:** Si BART detecta un nivel de riesgo, el trabajador debe ser evaluado con un equipo certificado antes de tomar cualquier decisión administrativa.

Esta clasificación está alineada con la posición de la OIT (Organización Internacional del Trabajo) sobre el uso de tecnologías de screening en ambientes laborales, que distingue entre:

- Herramientas diagnósticas: Requieren certificación y validez legal
- Herramientas preventivas: Enfoque educativo y de reducción de riesgos

2.4 Síntesis y Conclusiones del Estado del Arte

2.4.1 Hallazgos Principales

La revisión sistemática del contexto tecnológico, normativo y de mercado permite establecer las siguientes conclusiones:

1. **Brecha tecnológica validada:** Existe un espacio claramente definido entre los dispositivos certificados de alto costo y las soluciones académicas desconectadas. BART se posiciona como un puente funcional en este espacio, ofreciendo conectividad IoT a un costo accesible para instituciones educativas y empresas pequeñas.

2. **Viabilidad técnica del sensor MQ-3:** A pesar de sus limitaciones en selectividad y precisión absoluta, la literatura consultada (Hanwei Electronics, 2014; Fine et al., 2010) confirma que el sensor es adecuado para aplicaciones de screening

cualitativo, especialmente cuando se implementan técnicas de filtrado digital y calibración dinámica.

3. **Validación del stack tecnológico:** La combinación Arduino + Bluetooth + App Inventor ha sido utilizada exitosamente en múltiples proyectos IoT documentados, demostrando estabilidad y facilidad de integración. La principal innovación de BART radica en la arquitectura de comunicación bidireccional y el sistema de feedback multimodal.

4. **Marco legal favorable para dispositivos preventivos:** La Resolución 1843/2025 y la legislación sobre protección de datos en Colombia establecen un marco claro que permite el uso de dispositivos de screening siempre que se respeten los principios de confidencialidad y se utilicen con fines preventivos, no sancionatorios.

2.4.2 Justificación de Decisiones de Diseño

¿Por qué MQ-3 y no sensor electroquímico?

La decisión se fundamenta en tres factores:

1. **Restricción presupuestaria educativa:** Con un presupuesto de componentes de $120,000 COP, un sensor electroquímico ($180,000 COP solo el sensor) habría consumido el 150% del presupuesto total.

2. **Suficiencia funcional:** Para el propósito de sensibilización educativa, la precisión del MQ-3 (±0.05% BAC estimado) es adecuada para discriminar entre los tres estados de interés: sobrio, precaución y riesgo alto.

3. **Valor didáctico:** El trabajo con sensores de óxidos metálicos ofrece mayor riqueza pedagógica al requerir comprensión de divisores de voltaje,

acondicionamiento de señal y procesamiento digital, cumpliendo con los objetivos de aprendizaje de la asignatura de Computación Física.

¿Por qué Arduino UNO y no ESP32?

La elección de Arduino sobre ESP32 evita sobreingeniería por las siguientes razones:

1. **Conectividad innecesaria:** Las capacidades WiFi del ESP32 no aportan valor en un sistema de comunicación punto a punto local. El Bluetooth Classic del HC-05 es suficiente y consume menos energía.

2. **Complejidad de desarrollo:** El ESP32 requiere gestión de dos núcleos, configuración de FreeRTOS para multitarea, y manejo de un stack de red completo. Arduino UNO permite enfocarse en la lógica de negocio sin distracciones arquitecturales.

3. **Estabilidad probada:** El ecosistema Arduino tiene 18 años de madurez, con amplia documentación y comunidad de soporte. Para un proyecto académico con plazos definidos, esta estabilidad es crítica.

¿Por qué sistema de alimentación portátil con pilas AA y no fuente fija?

Una de las decisiones de diseño más significativas fue la implementación de un sistema de energía autónomo mediante pack de 6 pilas alcalinas AA (9V nominales regulados a 5V mediante el regulador de voltaje integrado del Arduino UNO).

Esta elección se fundamenta en tres pilares:

1. **Portabilidad operativa:** A diferencia de los dispositivos comerciales que requieren baterías recargables propietarias o alimentación por USB, el uso de pilas AA estándar permite:

   - Operación en entornos sin acceso a tomas eléctricas (puertas de acceso a plantas industriales, vehículos, zonas remotas)
   - Reemplazo inmediato de fuente de energía sin necesidad de períodos de carga
   - Disponibilidad universal del formato AA en cualquier comercio

2. **Autonomía extendida:** El consumo energético del sistema completo se estima en aproximadamente 250 mA promedio durante operación activa:

   - Arduino UNO: ~50 mA
   - Sensor MQ-3 (heater): ~150 mA
   - Módulo Bluetooth HC-05: ~40 mA (transmitiendo)
   - Barra de LEDs: ~10 mA (3 LEDs encendidos simultáneos)

Con un pack de pilas AA de 2500 mAh de capacidad típica, la autonomía teórica alcanza 10 horas de operación continua. En uso intermitente, la autonomía se extiende a varias semanas, cumpliendo con los requisitos de un dispositivo de screening que puede permanecer en un puesto de control sin mantenimiento frecuente.

3. **Simplicidad de mantenimiento:** En contextos de seguridad industrial, donde el personal operativo puede no tener conocimientos técnicos avanzados, la

capacidad de simplemente "cambiar las pilas" (como en un control remoto) reduce significativamente la barrera de uso y elimina dependencias de infraestructura de carga especializada.

Esta decisión contrasta con alternativas evaluadas:

- **Batería LiPo recargable:** Descartada por requerir circuito de carga especializado (TP4056), gestión de protección BMS (Battery Management System), y riesgos de seguridad en ambientes industriales (inflamabilidad).

- **Power bank USB:** Descartada por ocupar mayor volumen, introducir electrónica innecesaria (convertidores DC-DC), y agregar complejidad al diseño de la carcasa.

- **Alimentación por cable USB:** Contradice el objetivo de portabilidad total y limita severamente los casos de uso.

2.4.3 Sobre Diseño Gráfico Aplicado en el Proyecto

El diseño visual del sistema BART (tanto en la carcasa física como en el icono de la aplicación) adopta una estética Cyber-Industrial Design con influencias Y3K (Year 3000), buscando transmitir conceptos de innovación tecnológica, precisión técnica y contexto laboral industrial.

Fundamentos estéticos:

1. **Cyber-Industrial Design:** Esta corriente fusiona elementos de diseño industrial clásico (geometrías funcionales, estética de maquinaria) con visuales futuristas de ciencia ficción cyberpunk. Se caracteriza por:

- Uso de geometrías rígidas y angulares (hexágonos, biseles, cortes diagonales)
- Paleta cromática técnica: negro, gris metálico, acentos en cian/amarillo de advertencia
- Texturas que simulan materiales industriales (aluminio cepillado, plástico técnico)

2. **Estética Y3K:** Representa una evolución del diseño Y2K (año 2000), incorporando elementos más abstractos y digitales:

- Patrones de circuitos impresos y trazas electrónicas
- Fuentes tipográficas con carácter tecnológico (sans-serif geométricas, elementos monoespaciados)
- Elementos de UI que recuerdan interfaces HUD (Head-Up Display) de contextos militares o aeroespaciales

3. **Tipografía Cyber-Tech:** Se seleccionó una familia tipográfica con características técnicas específicas:

- Trazos uniformes que facilitan legibilidad en contextos de iluminación variable.
- Caracteres con proporciones inspiradas en fuentes de terminales de computadora.
- Esquinas biseladas que refuerzan la estética angular del diseño general.

4. **Técnicas visuales aplicadas:**

   - **Puntillismo técnico:** Uso de patrones de puntos degradados para crear sensación de profundidad y textura, emulando tramas de impresión industrial o mascarillas de PCB (Printed Circuit Board)

   - **Elementos geométricos abstractos:** Líneas de construcción, guías y marcos que refuerzan la sensación de precisión técnica y diseño asistido por computadora

Justificación contextual:

Para el público objetivo (supervisores de seguridad industrial, técnicos de SST, trabajadores en sectores de alto riesgo), el diseño busca transmitir:

1. **Seriedad técnica:** A pesar de ser un prototipo educativo, el tratamiento visual profesional genera credibilidad y respeto por parte de los usuarios.

2. **Diferenciación con dispositivos médicos tradicionales:** La estética cyber-industrial posiciona al dispositivo en el espacio de tecnología IoT innovadora, no como un "juguete" o dispositivo genérico.

3. **Coherencia con entornos industriales:** La paleta cromática de advertencia (amarillo, naranja, rojo) está alineada con los códigos visuales utilizados en señalización de seguridad industrial (Norma NTC 1461), facilitando la comprensión intuitiva de los estados del sistema.

4. **Identidad de marca educativa:** El diseño distintivo permite que el proyecto sea fácilmente identificable y recordable en contextos de presentación académica o ferias de ciencia.

2.4.4 Contribuciones Esperadas del Proyecto

Contribución Técnica

El proyecto BART aporta a la literatura técnica de sistemas embebidos IoT:

1. **Validación de arquitectura de bajo costo:** Documentación completa de una implementación funcional de telemetría biométrica con presupuesto inferior a $120,000 COP, demostrando que es posible crear sistemas IoT educativos sin comprometer funcionalidad básica.

2. **Protocolo de comunicación simplificado pero robusto:** La implementación de un esquema de comunicación bidireccional sobre Bluetooth SPP con sincronización basada en delimitadores ASCII demuestra que no siempre se requieren protocolos complejos (MQTT, CoAP) para aplicaciones de monitoreo en tiempo real.

3. **Técnicas de compensación por software:** El algoritmo de autocalibración dinámica y filtrado de promedio móvil implementado puede ser replicado en otros proyectos que utilicen sensores de óxidos metálicos, aportando valor más allá del dominio específico de alcoholimetría.

Contribución Aplicada:

Desde la perspectiva de Seguridad y Salud en el Trabajo:

1. **Herramienta de sensibilización accesible:** BART reduce la barrera de entrada para que pequeñas empresas puedan implementar programas de prevención de consumo de alcohol en contextos laborales, sin requerir inversiones de millones de pesos en equipos certificados.

2. **Registro histórico para auditorías:** La capacidad de almacenar mediciones con timestamps permite a las empresas documentar sus esfuerzos de prevención ante auditorías de ARL (Administradoras de Riesgos Laborales) o inspecciones del Ministerio del Trabajo.

3. **Promoción de cultura de autocontrol:** Al permitir que el trabajador acceda directamente a su información biométrica, se fomenta la responsabilidad individual y la conciencia sobre los riesgos del consumo de alcohol.

Contribución Académica:

Para la comunidad educativa en ingeniería:

1. **Caso de estudio integral:** El proyecto documenta un ciclo completo de desarrollo de sistema IoT: desde la conceptualización y revisión de estado del arte, hasta el diseño de hardware, implementación de firmware, desarrollo de aplicación móvil y validación experimental. Esto lo convierte en material didáctico valioso para cursos de computación física, sistemas embebidos o IoT.

2. **Código abierto:** Todo el código fuente están disponibles en el repositorio GitHub del proyecto.

3. **Metodología de validación:** La documentación del proceso de calibración, caracterización del sensor y pruebas de funcionamiento proporciona un framework metodológico que puede aplicarse a otros proyectos de instrumentación con sensores químicos.

2.4.5 Nexo con el Desarrollo del Proyecto

El análisis del estado del arte conduce directamente a la definición de requisitos funcionales y no funcionales del sistema:

Requisitos Funcionales Derivados

1. **RF-01:** El sistema debe permitir medición continua de concentración de alcohol en aire expirado con frecuencia mínima de 1 Hz.
    - Origen: Análisis de tiempos de respuesta en literatura de sensores MQ-3.
2. **RF-02:** El dispositivo debe proporcionar feedback visual local mediante indicador de barra progresivo.
    - Origen: Limitación identificada en proyectos académicos previos que dependían exclusivamente de conectividad móvil.
3. **RF-03:** La aplicación móvil debe clasificar el nivel de riesgo en tres categorías: Sobrio, Precaución y Riesgo Alto.
    - Origen: Estándares de la NHTSA y normativas de tránsito que establecen umbrales de 0.02%, 0.05% y 0.08% BAC.
4. **RF-04:** El sistema debe almacenar historial de mediciones con timestamps localmente en el dispositivo móvil.
    - Origen: Requisitos de la Ley 1581/2012 sobre protección de datos y soberanía de información del usuario.

Requisitos No Funcionales Derivados

1. **RNF-01:** El costo total de componentes no debe exceder $150,000 COP.

- Origen: Análisis de accesibilidad económica en mercado de dispositivos de screening.

2. **RNF-02:** La latencia entre la medición y la visualización en la aplicación no debe superar 2 segundos.

- Origen: Estudios de usabilidad en interfaces de monitoreo en tiempo real.

Arquitectura del Sistema Basada en Mejores Prácticas

Del análisis de proyectos similares, se adoptan las siguientes decisiones arquitectónicas:

1. **Separación de capas**: Implementación de arquitectura de tres capas (Sensórica, Procesamiento, Presentación) para facilitar mantenimiento y escalabilidad futura.

2. **Protocolo de handshake:** Implementación de comandos de control explícitos (Iniciar/Detener) inspirados en protocolos industriales como Modbus, garantizando control determinista del flujo de datos.

3. **Algoritmo de filtrado:** Selección de filtro de promedio móvil por su simplicidad de implementación y suficiencia para el nivel de ruido esperado, según literatura consultada.

Conclusión del Capítulo

El estado del arte confirma la viabilidad técnica y la pertinencia social del proyecto BART. La revisión sistemática de literatura, normativa y proyectos similares valida las decisiones de diseño tomadas y establece una base sólida de conocimiento sobre la cual se

construirá el sistema en los capítulos subsecuentes. La identificación clara del gap tecnológico (bajo costo + telemetría + cumplimiento normativo) posiciona al proyecto como una contribución original en el espacio de dispositivos IoT para salud ocupacional en contextos de recursos limitados.

3. Objetivos concretos y metodología de trabajo

Este capítulo establece la orientación del proyecto y define los objetivos que guían su desarrollo. A partir del análisis del contexto, del estado del arte y de las necesidades detectadas, se plantea un conjunto de metas que permiten estructurar el diseño, construcción y validación de un alcoholímetro portátil basado en Arduino y una aplicación móvil educativa.

Asimismo, se describe la metodología adoptada para abordar el proceso de creación, integración y evaluación del sistema.

3.1 Objetivo general

Desarrollar y validar un sistema portátil de medición de alcohol en el aliento, basado en Arduino y conectado a una aplicación móvil educativa mediante Bluetooth, que permita visualizar lecturas en tiempo real, registrar historial de consumo y fomentar la toma de decisiones responsables, demostrando la utilidad de la computación física para la prevención y el autocontrol personal.sos a seguir en el proceso de desarrollo del sistema software y las tecnologías a emplear.

3.2 Objetivos específicos

Para alcanzar el objetivo general, se establecen los siguientes objetivos específicos:

- Analizar los fundamentos teóricos asociados al consumo de alcohol, los principios de funcionamiento del sensor MQ-3 y las tecnologías de comunicación Bluetooth aplicadas a dispositivos portátiles.

- Describir el estado del arte sobre alcoholímetros educativos y prototipos basados en Arduino, identificando sus limitaciones y oportunidades de mejora.

- Diseñar la arquitectura del sistema, integrando hardware (MQ-3, Arduino, módulo HC-05 y LEDs) y software (aplicación móvil en App Inventor) en un único flujo de funcionamiento.

- Desarrollar el prototipo físico del alcoholímetro, programando el microcontrolador, calibrando el sensor y estableciendo la comunicación bidireccional con el dispositivo móvil.

- Implementar una aplicación móvil que permita visualizar las mediciones en tiempo real, almacenar datos, gestionar perfiles de usuario y calcular promedios de consumo.

- Verificar el funcionamiento del sistema mediante pruebas de lectura, comunicación, visualización y almacenamiento, evaluando su estabilidad, usabilidad y aporte educativo.

3.3 Metodología Del trabajo

3.3.1 Marco Metodológico General

Clasificación del proyecto: Desarrollo de prototipo tecnológico funcional con validación experimental en contexto de laboratorio académico.

Paradigma de desarrollo

El proyecto adoptó un enfoque de prototipado iterativo rápido inspirado en metodologías ágiles, adaptado a las restricciones de un proyecto académico de corta duración (3 semanas efectivas de desarrollo).

A diferencia de metodologías secuenciales tradicionales (modelo en cascada), donde cada fase se completa antes de iniciar la siguiente, este proyecto implementó ciclos de desarrollo concurrentes con retroalimentación continua.

Justificación del enfoque: La naturaleza del proyecto presentaba alta incertidumbre técnica en dos dimensiones críticas:

1. Trabajo con sensor discreto MQ-3 de 6 pines sin documentación estandarizada en español

2. Primera experiencia del equipo con desarrollo en MIT App Inventor y comunicación Bluetooth SPP

Estas condiciones hacían inviable un enfoque de planificación exhaustiva previa, requiriendo en su lugar un modelo de "aprendizaje por construcción" (learning by doing) donde los problemas técnicos se descubrían y resolvían durante la implementación.

Características del proceso aplicado

- Desarrollo concurrente hardware-software: El firmware de Arduino y la aplicación móvil se desarrollaron en paralelo, no de forma secuencial, permitiendo identificar problemas de integración tempranamente

- Debugging continuo como práctica de validación: Cada modificación del código se probaba inmediatamente en el hardware real, no solo en simulación

- Capacidad de pivotaje rápido: Ante problemas técnicos críticos (como el error de conexión del MQ-3), el equipo adoptó decisiones de corrección en menos de 24 horas, sin paralizar el proyecto

- Comunicación sincrónica del equipo: Sesiones diarias de trabajo conjunto (presencial y remoto) con intercambio inmediato de resultados de pruebas

Modelo de flujo general: El proceso siguió un ciclo iterativo de cuatro actividades fundamentales en cada fase:

1. Investigar, Consultar datasheets, tutoriales, proyectos similares.
2. Implementar, construir la funcionalidad específica.
3. Probar, Validar con casos de prueba reales en hardware.
4. Ajustar, Corregir errores y optimizar basándose en observaciones.

Este ciclo se repitió múltiples veces por día durante las fases críticas de desarrollo.

3.3.2 Fases del Desarrollo del Sistema

Fase 0: Conceptualización y Evaluación de Viabilidad

Duración: Semana 1 (5 días laborables)

Objetivo: Seleccionar una idea de proyecto técnicamente viable y relevante dentro del plazo académico establecido.

Actividades realizadas:

1. Propuesta inicial - Inhibidor de señal IoT: La primera idea explorada fue el desarrollo de un dispositivo capaz de bloquear selectivamente señales de radiofrecuencia en un área delimitada, con control remoto vía aplicación móvil. La motivación surgió del contexto de seguridad en exámenes académicos y espacios de concentración.

Durante la investigación preliminar se identificaron las siguientes barreras:

- Complejidad técnica: Los circuitos de jamming RF requieren conocimientos avanzados de electrónica de radiofrecuencia, modulación de señales y antenas direccionales, fuera del alcance de la asignatura

- Regulación legal: La Resolución CRC 5050 de 2016 de la Comisión de Regulación de Comunicaciones de Colombia prohíbe explícitamente la operación de

dispositivos inhibidores de señal sin autorización expresa del MinTIC, configurando su uso como delito

- Riesgo de cronograma: La probabilidad de no lograr un prototipo funcional en 4 semanas restantes se estimó superior al 60%

Decisión: Abandono de la idea por riesgos técnicos, legales y de viabilidad temporal.

2. Pivotaje hacia Alcoholímetro IoT:

La segunda propuesta surgió del análisis de problemáticas en seguridad y la reciente publicación de la Resolución 1843 del 2025 ministerio del Trabajo, que regula las pruebas de alcoholemia en entornos laborales.

Evaluación de viabilidad:

- Disponibilidad de componentes: Sensor MQ-3 disponible en mercado local (tiendas de electrónica en Neiva) con precio accesible (~$15,000 COP)
- Existencia de proyectos de referencia: Búsqueda en GitHub reveló múltiples implementaciones con Arduino, validando factibilidad técnica
- Relevancia social: Alineación con normativa vigente y problemática de seguridad laboral documentada
- Alcance controlado: Sistema con componentes bien definidos: sensor + microcontrolador + conectividad + app móvil

Criterio de selección final: Balance óptimo entre innovación técnica (integración IoT con telemetría), relevancia social (aplicación en SST) y probabilidad de éxito (tecnologías probadas).

Entregable de la fase: Documento de alcance del proyecto con justificación técnica y social, presentado al docente para aprobación.

Fase 1: Investigación Técnica Profunda

Duración: Semana 2

Objetivo: Adquirir conocimiento técnico especializado sobre las tecnologías a integrar mediante estudio de documentación oficial y análisis de proyectos de referencia.

Estudio del sensor MQ-3:

Fuentes consultadas:

- Datasheet oficial de Hanwei Electronics (versión 1.3, 8 páginas)
- Tutorial de Naylamp Mechatronics sobre sensores MQ-series
- Análisis de circuitos de aplicación en foros de Arduino

Conocimiento crítico adquirido:

- Comprensión de la arquitectura de 6 pines: separación entre circuito de calentamiento (H-H) y circuito de señal (A-B)
- Necesidad de resistencia pull-down de 10kΩ para estabilizar la señal analógica
- Tiempo de precalentamiento inicial: 24-48 segundos para estabilización completa
- Sensibilidad cruzada a otros VOCs: acetona, metano, compuestos aromáticos

Lección aprendida clave: La documentación de tutoriales en español mostraba mayoritariamente módulos pre-amplificados de 4 pines, generando confusión inicial sobre el

pinout del componente discreto. Esto subrayó la importancia de consultar datasheets oficiales del fabricante.

Estudio de arquitectura Arduino:

- Análisis del ADC de 10 bits:
  - Resolución: 1024 niveles sobre rango 0-5V = 4.88 mV/paso
  - Frecuencia de muestreo máxima: ~9600 Hz (más que suficiente para señales químicas con constante de tiempo en segundos)
- Investigación de SoftwareSerial:
  - Consulta del repositorio oficial en GitHub (PaulStoffregen/SoftwareSerial)
  - Comprensión de limitaciones: baudrate máximo confiable de 9600 bps, no recomendado para velocidades superiores
  - Ventaja identificada: Liberación del puerto USB para debugging simultáneo con transmisión Bluetooth

Estudio del protocolo Bluetooth SPP:

- Análisis del módulo HC-05:
  - Stack implementado: Bluetooth v2.0 + EDR en banda ISM 2.4 GHz
  - Perfil SPP (Serial Port Profile): Emulación de puerto serie virtual transparente sobre RFCOMM

- Configuración por defecto: 9600 baudios, 8N1 (8 bits de datos, sin paridad, 1 bit de parada)

- Integración con App Inventor:
  - Compatibilidad nativa del componente BluetoothClient con Bluetooth Classic (no BLE)
  - Necesidad de configurar DelimiterByte para sincronización de mensajes

Aprendizaje de desarrollo en App Inventor:

- Recursos utilizados:
  - Tutoriales oficiales de MIT sobre programación por bloques
  - Análisis de proyectos de ejemplo en galería de App Inventor
  - Tutorial específico: "Using MIT App Inventor to Control Arduino" (Instructables)
- Conceptos clave internalizados:
  - Paradigma de programación orientado a eventos (no secuencial)
  - Uso de componente Clock para polling periódico de datos
  - Manejo de almacenamiento local con TinyDB (base de datos key-value)

Metodología de aprendizaje: Combinación de learning by reading (estudio de documentación) y benchmarking (análisis de código de proyectos open-source similares en GitHub).

Entregables de la fase:

- Esquemático preliminar del circuito dibujado
- Pseudocódigo de lógica de control del firmware

Fase 2: Implementación de Hardware y Debugging Crítico

Duración: Días 1-3 de Semana 3

Objetivo: Construir el circuito físico funcional y validar comunicación entre todos los componentes.

Actividades de construcción:

Montaje en protoboard:

Disposición de componentes siguiendo esquema preliminar

Integración del pack de 6 pilas AA (9V nominal) conectado al jack barrel del Arduino UNO, regulado internamente a 5V

Conexión de barra de 8 LEDs en pines digitales 5-12, cada uno con resistencia limitadora de 220Ω en serie

Cableado del módulo Bluetooth HC-05:

VCC, 5V del Arduino

GND, GND común

TX del HC-05, Pin 2 del Arduino (configurado como RX en software)

RX del HC-05, Pin 4 del Arduino (configurado como TX en software)

Primera prueba de subsistemas:

- Barra de LEDs: Funcionamiento correcto al primer intento. Se cargó un sketch de prueba (barrido secuencial tipo "Knight Rider") que encendió los 8 LEDs progresivamente, validando conectividad y polaridad correcta

- Módulo Bluetooth: Emparejamiento exitoso desde smartphone. El módulo aparecía en la lista de dispositivos disponibles con nombre "HC-05" y se enlazaba correctamente introduciendo PIN 1234

Problema crítico identificado: Error de conexión del sensor MQ-3

Síntoma observado

Al conectar el sensor MQ-3 según el esquema inicial y ejecutar lecturas del pin A0, se obtenían valores extremadamente bajos y estables (30-70 en la escala ADC de 0-1023) en

aire limpio, sin variación significativa al exponer el sensor a vapores de alcohol isopropílico. Los LEDs permanecían apagados o encendían solo el primero.

Proceso de diagnóstico

1. Verificación de alimentación: Se confirmó con multímetro que el sensor recibía 5V en sus terminales de alimentación (voltaje correcto)

2. Revisión del código: Se simplificó el código a una lectura básica de analogRead(A0) con impresión en Monitor Serie, descartando problemas de software

3. Análisis del circuito físico: Se rastreó la conexión del pin A0 y se descubrió que estaba conectado directamente a uno de los pines del heater (pines centrales H), no al pin de señal

Comprensión del error

El circuito calefactor del sensor MQ-3 (pines H-H) tiene como única función mantener el elemento de SnO_2 a ~350°C para activar la reacción química. Este circuito tiene baja resistencia (~33Ω) y al conectarlo al ADC a través de una resistencia pull-down, se creaba un

divisor de voltaje que entregaba valores muy bajos al pin A0. Adicionalmente, lo que se estaba midiendo era la temperatura del heater, no la resistencia variable del elemento sensor.

Proceso de corrección

1. Consulta exhaustiva del datasheet original: Se identificó en la página 2 del documento de Hanwei Electronics la descripción clara de los dos circuitos independientes:

   - Heater (H-H): Pines centrales (4 y 5), conectados directamente a 5V y GND
   - Sensor (A-B): Pines externos (1-2 y 3-6), configurados como divisor de voltaje con A a 5V y B al ADC con pull-down de 10kΩ a GND

2. **Recableado del circuito**

   - Pin A del sensor (esquina superior izquierda), 5V
   - Pin B del sensor (esquina inferior izquierda), Pin A0 del Arduino
   - Resistencia de 10kΩ entre pin A0 y GND (pull-down)
   - Pines H-H mantuvieron conexión directa a alimentación

3. Precaución adicional: Se monitoreó la temperatura superficial del sensor durante 10 minutos para verificar que no se había dañado por el error previo (temperatura estable ~40°C, sin humo ni olor, indicando integridad del componente)

Resultado post-corrección

Tras el recableado, el sensor comenzó a responder correctamente:

- Aire limpio: Lecturas estables en rango 180-220 ADC
- Vapores de alcohol (algodón embebido en alcohol isopropílico a 10 cm): Picos de 500-750 ADC
- Barra de LEDs: Respuesta progresiva correspondiente a la concentración, validando toda la cadena de procesamiento

Validación de otros subsistemas

- Comunicación Bluetooth: Enlace establecido, pero se identificó conflicto de pines (documentado en Fase 3)
- Sistema de alimentación: Voltaje estable de 5V en todos los componentes incluso con carga completa (sensor + LEDs + Bluetooth)

Criterios de aceptación alcanzados

- Sensor MQ-3 responde diferencialmente a presencia de alcohol (Delta >300 unidades ADC)
- LEDs responden individualmente a comandos digitalWrite() sin parpadeos ni falsos contactos
- Módulo HC-05 emparejable desde smartphone Android sin necesidad de apps de terceros
- Comunicación UART con HC-05 requiere solución alternativa (Fase 3)

Lección crítica aprendida

La información de tutoriales en internet puede ser imprecisa o aplicarse a versiones diferentes de componentes. En electrónica, la fuente de verdad absoluta es el datasheet del fabricante.

Adicionalmente, ante un comportamiento anómalo, la verificación física del circuito con multímetro (continuidad, voltajes) debe ser el primer paso antes de asumir problemas de código.

Fase 3: Desarrollo Integrado de Firmware y Aplicación Móvil

Duración: Semana 3 (3 dias intensivos)

Objetivo: Implementar la lógica de control completa en Arduino y desarrollar la interfaz de usuario en App Inventor, estableciendo un canal de comunicación bidireccional estable.

Metodología de programación: Desarrollo incremental con integración continua hardware-software. Cada funcionalidad nueva se probaba inmediatamente en el sistema completo antes de avanzar a la siguiente.

Resolución del problema de comunicación Bluetooth

Contexto del problema

Los pines TX (1) y RX (0) del Arduino UNO están físicamente conectados al chip convertidor USB-Serial (ATMEGA16U2 o CH340 según revisión de placa).

Esta conexión permite cargar código desde el IDE y usar el Monitor Serie. Sin embargo, al conectar el módulo Bluetooth a estos mismos pines se genera un conflicto:

- Al cargar código, el Bluetooth recibe bytes espurios del proceso de programación
- Al usar el Monitor Serie, los datos del Bluetooth se mezclan con los comandos de depuración
- El resultado es comunicación inestable e impredecible

Solución implementada: SoftwareSerial en pines alternativos

Se utilizó la librería SoftwareSerial.h para crear un puerto serie virtual en pines digitales no comprometidos:

```
30  #include <SoftwareSerial.h>
31
32  // --- Pines Config ---
33  const int PIN_RX_BT = 2;     // TX del Bluetooth
34  const int PIN_TX_BT = 4;     // RX del Bluetooth
35
36  // --- Obj ---
37  SoftwareSerial BTSerial(PIN_RX_BT, PIN_TX_BT);
38
39  void setup() {
40    // 1. Inicialización de Puertos
41    Serial.begin(9600);
42    BTSerial.begin(9600);    // Telemetría
```

Figura 1: Extracto de código desarrollado para configurar la comunicación serial alternativa (SoftwareSerial) y permitir el debugging simultáneo.

Ventaja lograda: Debugging simultáneo

Durante el desarrollo, se podía observar en el Monitor Serie (puerto USB) los valores internos del sensor mientras simultáneamente se transmitían datos clasificados al smartphone vía Bluetooth.

Esto aceleró dramáticamente el ciclo de desarrollo, reduciendo el tiempo de identificación de errores de horas a minutos.

Trade-off aceptado: La librería SoftwareSerial no es confiable a velocidades superiores a 9600 baudios. Sin embargo, para este proyecto que transmite únicamente 1 byte por segundo (valores 1, 2 o 3), esta limitación no representa un problema.

Iteraciones de desarrollo del firmware

El firmware se construyó mediante 5 iteraciones incrementales:

Iteración 1

Lectura básica y calibración automática

- Implementación de rutina en setup() que captura 20 muestras del sensor en aire limpio
- Cálculo del promedio aritmético almacenado en variable global baseAireLimpio
- Propósito: Compensar variaciones de temperatura, humedad y altitud del entorno
- Validación: Se ejecutaron pruebas en tres momentos del día (mañana, mediodía, tarde) verificando que la línea base se ajustaba automáticamente (variación de ±15 unidades ADC entre sesiones)

Iteración 2: Filtro de promedio móvil

- Implementación de función leerPromedioSensor(int muestras) que toma N lecturas consecutivas con delay de 4 ms entre cada una
- Justificación del delay: El ADC del ATmega328P requiere ~100 µs por conversión, pero se añade tiempo adicional para estabilización del multiplexor interno
- Resultado medido: Reducción del ruido de lectura de ±15 unidades ADC (lectura directa) a ±3 unidades (promedio de 10 muestras)

Iteración 3: Mapeo de niveles y control de LEDs

- Conversión de valor delta (0-350 ADC) a niveles de LEDs (0-8) usando función map() de Arduino

- Implementación de mostrarBarraLeds(int nivel) que enciende progresivamente los LEDs hasta el nivel indicado

- Problema detectado: En nivel 0, el usuario no percibía visualmente si el sistema estaba midiendo o apagado

- Solución: Modificación del código para mantener el primer LED verde encendido permanentemente cuando pruebaActiva == true, funcionando como "heartbeat" del sistema

Iteración 4: Traducción semántica para la App

- Categorización de 8 niveles de hardware en 3 niveles de software según rangos fisiológicos:

```
129         if (nivelLed <= 3) {
130           nivelApp = 1; // Zona Segura (Verde)
131         }
132         else if (nivelLed > 3 && nivelLed <= 6) {
133           nivelApp = 2; // Zona Precaución (Amarillo)
134         }
135         else if (nivelLed > 6) {
136           nivelApp = 3; // Zona Peligro (Rojo)
137         }
```

Figura 2: Fragmento de código de Arduino

- Transmisión vía Bluetooth: BTSerial.println(nivelApp); envía "1\n", "2\n" o "3\n" (carácter de nueva línea como delimitador)

Iteración 5: Protocolo de comandos bidireccional

- Implementación de función verificarComandosEntrantes() que escucha tanto Serial como BTSerial
- Comandos definidos:
  - 'I' o '1': Iniciar medición (recalibra línea base y activa flag pruebaActiva)
  - 'X' o '2': Detener medición (desactiva flag y apaga todos los LEDs)
  - 'S': Guardar medición (feedback visual con parpadeo completo de barra LED)

- Animaciones de feedback: Cada comando recibido genera un pulso breve en el LED integrado del pin 13, confirmando recepción

Desarrollo paralelo de la aplicación móvil

Pantalla 1: Gestión de conexión Bluetooth

- Componente ListPicker (selector de lista) que ejecuta en su evento BeforePicking la función BluetoothClient.AddressesAndNames para cargar dispositivos emparejados

- Lógica en evento AfterPicking:

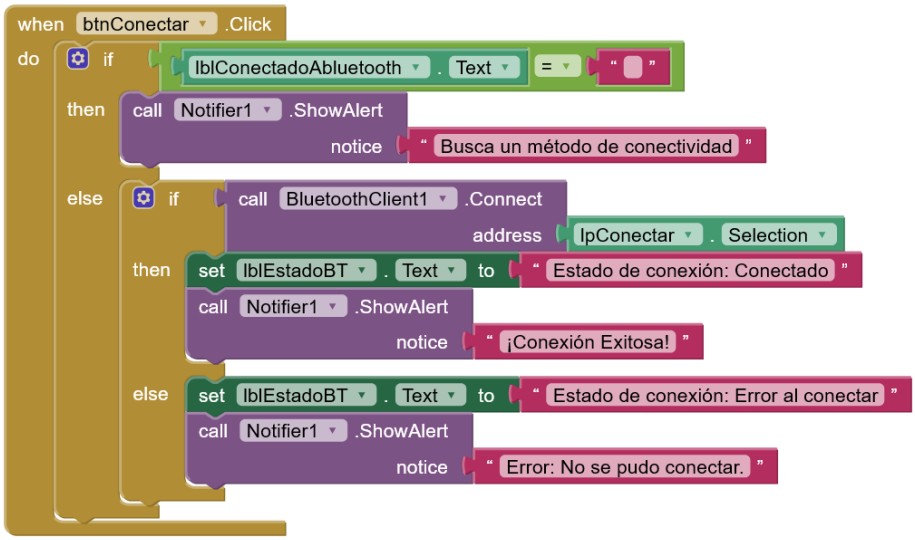


Figura 3: Fragmento de código de App Inventor

- Manejo de error: Verificación del estado de conexión antes de cambiar de pantalla

Pantalla 2: Monitoreo en tiempo real

- Componente Clock configurado con TimerInterval = 500 (polling cada 500 ms)

- Lógica del evento Clock.Timer:

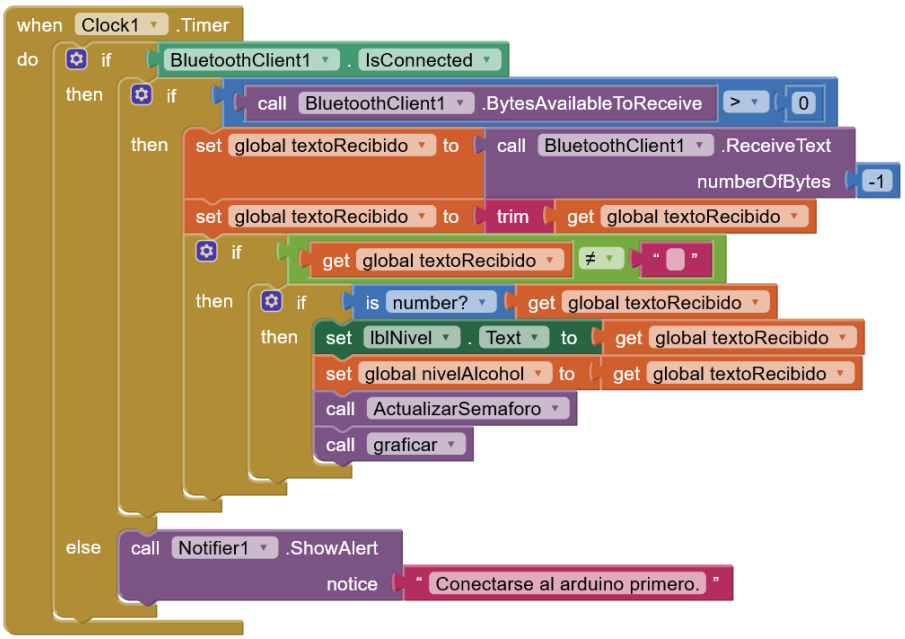


Figura 4: Fragmento de código de App Inventor

(El parámetro -1 indica leer hasta encontrar el delimitador configurado)

Problema crítico detectado: Concatenación de datos

Síntoma: La variable nivelAlcohol contenía valores concatenados como "123" o "[0 0]" en lugar de valores individuales actualizados.

Causa raíz: Por defecto, App Inventor no interpreta saltos de línea como finalizadores de mensaje. Sin configurar el DelimiterByte, el componente BluetoothClient acumula todos los bytes recibidos hasta que se invoca ReceiveText, resultando en concatenación.

Solución implementada: En el evento Screen2.Initialize:

Resultado: Cada llamada a ReceiveText ahora devuelve un único valor ("1", "2" o "3"), sincronizado correctamente con las transmisiones del Arduino.

Sistema de semáforo visual:

- Se diseñaron 3 elementos canvas (verde, amarillo, rojo)
- Lógica condicional para cambio dinámico:

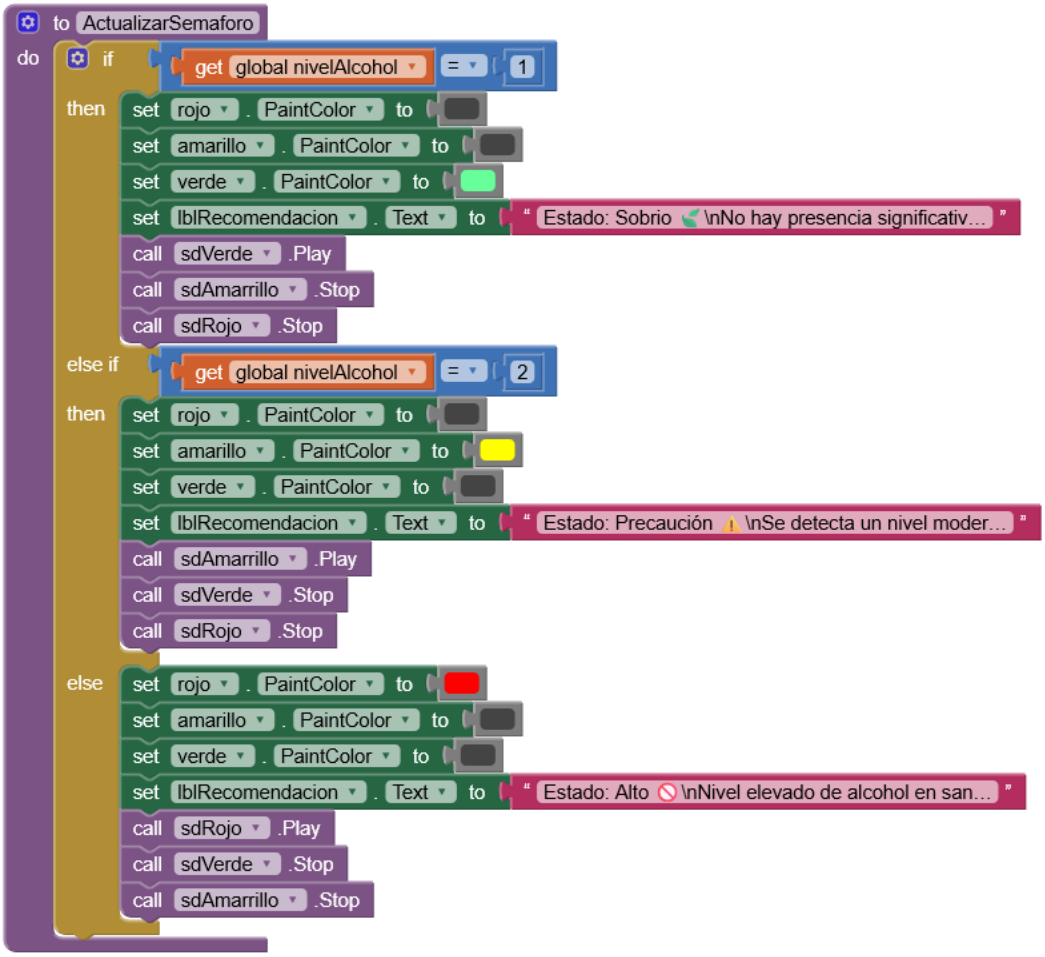


Figura 5: Fragmento de código de App Inventor

Módulo de historial con persistencia local:

- Botón "Guardar Medición": Al presionarse, almacena en TinyDB el valor actual con un timestamp:

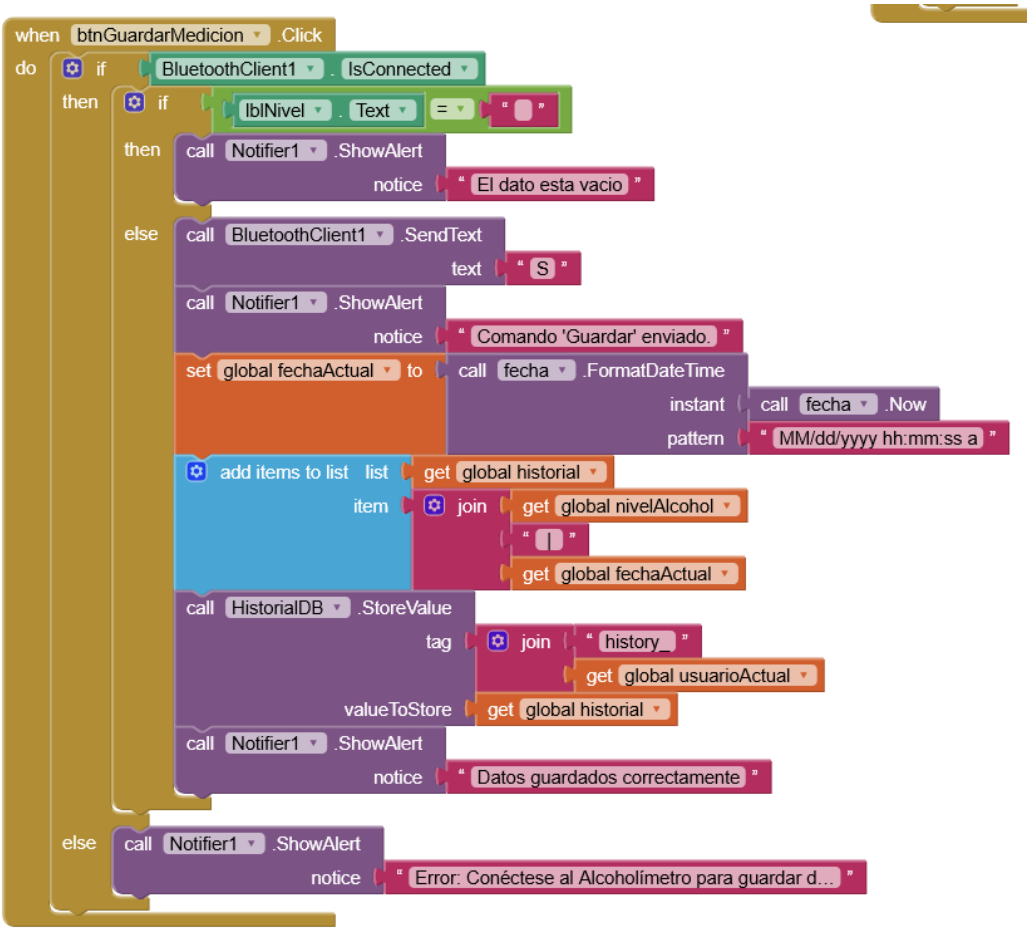


Figura 6: Fragmento de código de App Inventor

- Pantalla de historial: Recupera todas las entradas usando TinyDB.GetEntries y las muestra en un ListView con formato legible

Proceso de validación cruzada

Metodología de trabajo en equipo:

- Desarrollo paralelo: Mientras un integrante modificaba el firmware Arduino, el otro ajustaba la interfaz de App Inventor

- Sincronización continua: Uso de grupo de WhatsApp para compartir videos cortos (15-30 segundos) de pruebas en tiempo real

- Debugging colaborativo: Sesiones de 2-3 horas en un lugar físico donde ambos revisabamos el código simultáneamente

Ciclo típico de iteración:

1. Modificación de código (Arduino o App Inventor)
2. Carga/compilación (5 minutos)
3. Prueba funcional (7 minutos)
4. Análisis de resultados en equipo (5 minutos)
5. Decisión: continuar o ajustar (3 minuto)

Este ciclo se repetía 8-10 veces por sesión de trabajo, permitiendo avance rápido mediante retroalimentación inmediata.

Fase 4: Ensamblaje Final y Optimización Física

Duración: Ultimos días de la semana 3

Objetivo: Transformar el prototipo de breadboard en un dispositivo portable y robusto mediante integración en carcasa con sistema de alimentación autónomo.

Implementación del sistema de alimentación portátil

Configuración adoptada:

Pack de 6 pilas alcalinas AA conectadas en serie:

- Voltaje nominal: 6 × 1.5V = 9V
- Capacidad típica: 2500 mAh por celda

- Regulación: El Arduino UNO integra un regulador de voltaje (LM7805 o similar) que convierte 7-12V de entrada DC a 5V estabilizados para el microcontrolador y periféricos

Cálculo de autonomía teórica:

- Consumo promedio del sistema completo: ~250 mA
    - Arduino UNO: ~50 mA
    - Sensor MQ-3 (heater): ~150 mA
    - Módulo HC-05: ~40 mA (transmitiendo)
    - Barra de 8 LEDs: ~10 mA (3 LEDs encendidos simultáneos)

En uso intermitente real (mediciones de 5 minutos cada 2 horas), esta autonomía se extiende a varias semanas.

Ventajas de la solución de pilas AA:

- Disponibilidad universal: Formato estándar disponible en cualquier tienda de conveniencia
- Reemplazo instantáneo: Sin necesidad de períodos de carga (crítico en entornos operativos)
- Costo operativo bajo: Pilas recargables NiMH (~$3,000 COP/3 unidades) con 500 ciclos de vida útil
- Simplicidad de mantenimiento: El usuario final puede cambiar las pilas sin asistencia técnica

Conexión implementada:

Pack de pilas, Jack barrel 2.1mm, entrada DC del Arduino UNO

Validación: Prueba de operación continua de 2 horas sin degradación observable de la señal del sensor ni apagados espontáneos.

Diseño y construcción de carcasa

Material: Cartón corrugado de doble pared (grosor 5mm), seleccionado por disponibilidad, bajo costo y facilidad de procesamiento manual.

Dimensiones aproximadas: 15.25cm (largo) × 20.1cm (ancho) × 4.15cm (alto)

Características estructurales implementadas:

1. Ventana frontal para barra LED: Apertura rectangular (8 cm × 2 cm) con cubierta transparente para protección contra polvo manteniendo visibilidad

2. Abertura superior con rejilla: Orificio circular (diámetro 2.5 cm) directamente sobre el sensor MQ-3, permitiendo flujo de aire. Cubierto con malla metálica fina para evitar entrada de objetos extraños

3. Orificio posterior para USB y fuente: Permite conectar cable de programación sin desarmar, facilitando actualizaciones de firmware

Proceso de construcción:

1. Diseño de plantilla en papel milimetrado

2. Transferencia de medidas al cartón con lápiz

3. Corte con cutter de precisión guiado por regla metálica

4. Ensamblaje reforzado con cinta adhesiva

Acabado visual:

- Aplicación de diseño gráfico impreso en papel adhesivo fotográfico
- Estética cyber-industrial con tipografía técnica y elementos geométricos (ver sección 2.4.3 del documento)

Miniaturización de componentes pasivos

Optimización de resistencias:

- Las resistencias comerciales tipo axial tienen terminales de 25-30 mm de longitud total
- Mediante cortafrio de precisión, se recortaron los terminales a 8-10 mm de cada lado
- Esta reducción de 60% en longitud permitió:
    - Montaje más compacto en el protoboard (menor espacio entre filas)
    - Mejor estética visual del ensamblaje interno
    - Reducción de riesgo de cortocircuitos por contacto accidental con terminales largos

Precaución tomada: Se verificó con multímetro que el valor de resistencia no cambiara después del corte (variación <1%, dentro de tolerancia del 5% de las resistencias).

Gestión térmica pasiva

Posicionamiento del sensor MQ-3:

- Ubicación en la zona lateral de la carcasa, directamente bajo la abertura de ventilación
- Justificación: El calor generado (~0.9W) tiende a ascender por convección natural, facilitando su evacuación

Sistema de ventilación:

- Efecto chimenea: Flujo de aire por diferencia de temperatura sin necesidad de ventilación forzada

Validación térmica:

- Medición de temperatura superficial de la carcasa con termómetro infrarrojo después de 30 minutos de operación continua: 34°C (temperatura ambiente: 29°C)
- Delta térmico de 9°C considerado aceptable para componentes electrónicos (temperatura de juntura de semiconductores <50°C)

Pruebas de robustez mecánica

Test de impacto:

- Procedimiento: Caída libre desde altura de 50 cm sobre superficie de concreto
- Resultado: Sin desconexiones de cables, sin fisuras visibles en soldaduras, funcionamiento normal al encender
- Conclusión: Nivel de protección suficiente para transporte en bolso o mochila

Test de vibración:

- Procedimiento: Transporte del dispositivo en mochila durante caminata de 1 hora en terreno urbano irregular
- Verificación posterior: Medición de lectura del sensor (sin cambios en línea base), prueba de todos los LEDs (encendido correcto), prueba de conexión Bluetooth (enlace establecido sin errores)
- Conclusión: Sistema confiable para uso móvil en campo

Entregables de la fase:

- Dispositivo completamente ensamblado y operativo

Fase 5: Validación Experimental y Ajuste Fino

Duración: Ultimo dia

Objetivo: Caracterizar el comportamiento del sistema bajo condiciones controladas, ejecutar protocolo de calibración y optimizar parámetros operativos.

Actividades ejecutadas:

1. Ejecución del protocolo de calibración (ver sección 3.3.3.1) con 5 repeticiones de medición usando vapores de alcohol isopropílico
2. Pruebas de conectividad Bluetooth (ver sección 3.3.3.2) evaluando rango efectivo y latencia de comunicación
3. Evaluación de usabilidad con 3 usuarios piloto para identificar problemas de interfaz
4. Ajuste de parámetros del firmware basado en observaciones empíricas:

- Incremento del RANGO_DETECCION de 300 a 350 unidades ADC (sensor mostraba saturación prematura en pruebas con alta concentración)
- Modificación del INTERVALO_ENVIO de 500ms a 1000ms para reducir tráfico Bluetooth (la actualización cada medio segundo resultaba excesiva para la velocidad de cambio de la señal química)

Escenarios de prueba ejecutados:

Test 1: Estabilidad de línea base en diferentes ambientes

- Mediciones en 3 ubicaciones: interior de vivienda, exterior (patio), laboratorio de la universidad
- Resultado: Variación de línea base de ±18 unidades ADC entre ambientes (rango aceptable)
- Conclusión: El algoritmo de autocalibración compensa exitosamente diferencias ambientales

Test 2: Curva de respuesta vs. distancia de exposición

- Vapores de alcohol isopropílico presentados a distancias de 5, 10 y 15 cm del sensor
- Lecturas promedio (5 muestras por distancia):
  - 5 cm: 680 ADC (nivel 7-8 de LEDs)
  - 10 cm: 420 ADC (nivel 5 de LEDs)
  - 15 cm: 280 ADC (nivel 3-4 de LEDs)

- Conclusión: Respuesta inversamente proporcional a distancia, comportamiento esperado según ley del cuadrado inverso de difusión de gases

Test 3: Deriva térmica de largo plazo

- Operación continua durante 4 horas con registro de línea base cada 30 minutos
- Resultado: Deriva acumulada de +25 unidades ADC en 4 horas (incremento del 12% respecto a valor inicial)
- Interpretación: Efecto de calentamiento progresivo del ambiente interno de la carcasa
- Medida correctiva: Recomendación de recalibración cada 2 horas en uso prolongado

Test 4: Interferencia electromagnética

- Exposición a fuentes comunes de EMI:
    - Router WiFi 2.4GHz a 20 cm: Sin efectos observables
    - Smartphone realizando llamada a 10 cm: Sin efectos observables
- Conclusión: Señal analógica del sensor suficientemente robusta, sin necesidad de blindaje adicional

Ajustes post-validación

Justificación: En pruebas con alcohol isopropílico puro (99%), el sensor alcanzaba valores de 800+ ADC, pero el mapeo a 8 LEDs saturaba en 300. El nuevo valor permite mejor resolución en el extremo alto.

Optimización 2: Frecuencia de transmisión

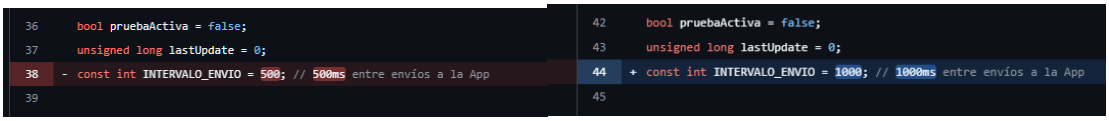


Figura 7: Fragmento de código de Arduino

Justificación

La señal del sensor MQ-3 tiene constante de tiempo en el orden de segundos. Enviar datos cada 0.5s generaba tráfico redundante sin aportar información nueva. La reducción a 1s

mejora eficiencia energética (menor actividad del transmisor Bluetooth) sin sacrificar responsividad percibida.

Optimización 3: Feedback visual de estado "Listo"

- Modificación de la animación de inicio: parpadeo de LEDs verdes de 3 a 5 ciclos
- Justificación: En pruebas de usabilidad, 2 usuarios no percibieron la señal de "sistema listo" con solo 3 parpadeos rápidos

3.3.3. Metodología de Validación Técnica

3.3.3.1. Protocolo de Calibración del Sensor

Objetivo: Establecer línea base de referencia y caracterizar la respuesta del sensor MQ-3 bajo condiciones controladas.

Materiales necesarios:

- Alcohol isopropílico 70% (disponible en farmacias)
- Algodón absorbente (bolas de algodón estéril)
- Contenedor hermético transparente de ~500 ml (frasco de vidrio con tapa de rosca)
- Cronómetro (aplicación de smartphone)

Procedimiento detallado:

Paso 1: Estabilización térmica del sensor

- Encender el dispositivo y esperar 30-60 segundos

- Verificación cualitativa: Acercar la mano a 2 cm del sensor (sin tocar) para percibir calor radiante del heater
- Observación de LEDs: El primer LED verde debe estar encendido indicando sistema activo

Paso 2: Medición de línea base (aire limpio)

- Exponer el sensor a aire ambiente de interior sin fuentes de VOCs (evitar proximidad a perfumes, desinfectantes, productos de limpieza)
- Registrar 20 muestras consecutivas con intervalo de 5 segundos cada una
- Anotar cada valor mostrado en el Monitor Serie del Arduino IDE
- Cálculo: $\sqrt{\frac{1}{19}\sum_{i=1}^{20}(\text{Muestra}_i - \text{Línea Base})^2}$

Valor de referencia esperado: 200-250 unidades ADC para aire a 25°C y 60% de humedad relativa (según datasheet de Hanwei Electronics).

Paso 3: Generación de atmósfera de prueba

- Aplicar 2-3 ml de alcohol isopropílico sobre una bola de algodón usando jeringa o gotero
- Introducir el algodón en el fondo del contenedor hermético
- Cerrar la tapa y agitar suavemente durante 10 segundos
- Esperar 30-60 segundos adicionales para permitir saturación del aire dentro del contenedor

- Indicador visual: Aparición de condensación (gotas microscópicas) en las paredes internas del frasco

Paso 4: Medición de respuesta dinámica

- Abrir la tapa del contenedor y aproximar el sensor a la boca del frasco (distancia ~5 cm)
- Activar el cronómetro al momento de la aproximación
- Registrar las siguientes métricas:
    - Tiempo de onset (t_onset): Momento en que la lectura supera Línea Base + 20 unidades ADC
    - Tiempo de pico (t_peak): Momento en que la lectura alcanza su valor máximo
    - Valor de pico (V_peak): Lectura máxima registrada en unidades ADC
    - Tiempo de recuperación (t_recovery): Momento en que la lectura retorna a Línea Base + 10%
- Retirar el sensor del contenedor cuando se alcance el pico

Paso 5: Evaluación de repetibilidad

- Realizar 5 mediciones consecutivas siguiendo el procedimiento del Paso 4
- Entre cada medición, esperar tiempo de recuperación completo (típicamente 3-5 minutos)

- Verificación visual: La barra de LEDs debe retornar a nivel 0 o 1 antes de iniciar nueva medición

Análisis de datos:

- Cálculo del coeficiente de variación para el valor de pico:

$$CV\% = \frac{\text{Desviación Estándar}}{\text{Promedio}} \times 100$$

- Generación manual de gráfico curva de respuesta típica (eje X: tiempo en segundos, eje Y: lectura ADC)

Criterios de aceptación para sensor funcional:

- CV% < 20% (tolerancia ampliada considerando naturaleza de sensor de bajo costo)
- Tiempo de respuesta (t_peak) < 15 segundos
- Tiempo de recuperación (t_recovery) < 90 segundos
- Incremento mínimo: V_peak > Línea Base + 100 unidades ADC

Interpretación de resultados:

- Si CV% > 20%: Posible contaminación del sensor o flujo de aire irregular. Repetir prueba en ambiente más controlado
- Si t_peak > 15s: Sensor podría estar al final de su vida útil (degradación del elemento SnO_2) o temperatura de heater insuficiente
- Si incremento < 100 ADC: Concentración de alcohol insuficiente en contenedor o error en conexión del circuito

3.3.3.2. Pruebas de Conectividad y Telemetría

Objetivo: Validar la estabilidad del enlace Bluetooth SPP y la integridad de los datos transmitidos entre Arduino y aplicación móvil.

Prueba 1: Rango efectivo de operación

Procedimiento:

1. Establecer conexión Bluetooth desde la aplicación móvil
2. Enviar comando 'I' para iniciar transmisión continua de datos
3. Posicionar el dispositivo Arduino en un punto fijo (ej: mesa)
4. Alejar progresivamente el smartphone en línea recta con incrementos de 1 metro
5. En cada distancia, registrar:
    - Estado de conexión en la app (conectado / desconectado)
    - Número de paquetes perdidos en una muestra de 20 transmisiones (observar si la imagen del semáforo se actualiza 20 veces o menos)

Variantes de la prueba:

- Escenario A (línea de vista): Sin obstáculos entre dispositivos
- Escenario B (con obstáculo): Colocar una pared de concreto o puerta cerrada entre dispositivos

Valores de referencia (según especificaciones Bluetooth Clase 2):

- Línea de vista: 8-10 metros sin pérdida de conexión
- Con obstáculo (pared de concreto de 15 cm): 3-5 metros

Registro de datos:

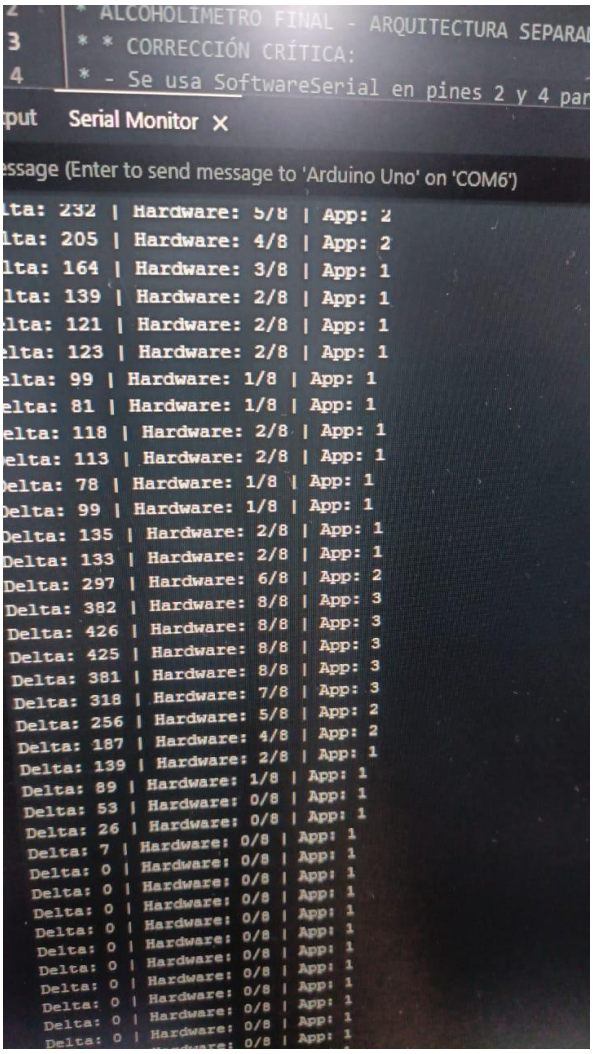


Figura 8: Terminal de Monitor Serial en Arduino IDE evidenciando los datos de prueba

Prueba 2: Latencia de comunicación (evaluación cualitativa)

Procedimiento simplificado:

1. Exponer el sensor a vapores de alcohol para generar un cambio de nivel detectable

2. Observar simultáneamente:

    - Barra de LEDs en el dispositivo físico
    - Imagen del semáforo en la pantalla de la aplicación móvil

3. Estimar cualitativamente el desfase temporal entre ambas visualizaciones

Criterio de aceptación cualitativo:

- Latencia imperceptible para el usuario (<500 ms, es decir, ambas visualizaciones parecen cambiar "al mismo tiempo")
- Latencia perceptible pero aceptable (500-1000 ms, cambio en app ocurre "un instante después")
- Latencia inaceptable (>1000 ms, retraso obvio y molesto para el usuario)

La latencia real del stack Bluetooth SPP es típicamente 40-80 ms. El Clock.Timer de App Inventor configurado a 500 ms introduce latencia adicional (polling). La latencia total percibida es la suma de ambos factores.

Prueba 3: Integridad de datos

Procedimiento:

1. Modificar temporalmente el código Arduino para transmitir una secuencia predefinida
2. Observar en la aplicación móvil la secuencia recibida
3. Registrar cualquier anomalía:

- Valores incorrectos (ej: se envió "2" pero se recibió "3")
- Valores duplicados (ej: se recibió "2 2" en lugar de "2")
- Valores perdidos (ej: no se actualizó la imagen del semáforo en un ciclo)

Meta de calidad:

- Tasa de error < 1% (máximo 1 error en 100 transmisiones)

- Si tasa de error > 1%: Posible interferencia electromagnética, baudrate mal configurado, o problema en configuración de DelimiterByte

Prueba 4: Estabilidad de conexión prolongada

Procedimiento:

1. Establecer conexión Bluetooth
2. Iniciar transmisión continua (comando 'I')
3. Dejar el sistema operando sin intervención durante 1 hora
4. Registrar cada desconexión espontánea (observable por mensaje de error en la app o detención de actualizaciones)
5. Al finalizar, verificar que el enlace siga activo

Meta: Cero desconexiones espontáneas en condiciones de operación sin interferencias severas.

Causas posibles de desconexiones:

- Agotamiento de batería del smartphone (verificar nivel de carga >50% al inicio)
- Interferencia de otros dispositivos Bluetooth en proximidad
- Timeout del sistema operativo Android (algunos fabricantes implementan políticas agresivas de ahorro de energía que desconectan Bluetooth si no hay interacción del usuario)

3.3.4. Gestión de Riesgos del Proyecto

El proyecto implementó un proceso simplificado de gestión de riesgos inspirado en ISO 31000:2018, consistente en identificación temprana, análisis de probabilidad/impacto, y definición de estrategias de mitigación.

Matriz de Riesgos Identificados:

ID	Descripción del Riesgo	Probabilidad	Impacto	Nivel de Riesgo	Estrategia de Mitigación	Estado Final
R-01	Sensor MQ-3 defectuoso o sin respuesta inicial	Media	Alto	**ALTO**	Adquirir 2 unidades del sensor; validar continuidad con multímetro antes de integrar	MATERIALIZADO Y RESUELTO (error de conexión, no defecto)
R-02	Incompatibilidad Bluetooth HC-05 con componentes de App Inventor	Baja	Alto	**MEDIO**	Validación temprana en Fase 2; tener módulo HC-06 como respaldo	MITIGADO (SoftwareSerial resolvió el problema)
R-03	Deriva excesiva del sensor por temperatura/humedad ambiente	Alta	Medio	**ALTO**	Implementar recalibración manual por comando; documentar limitaciones en manual de usuario	MITIGADO (recalibración con comando 'I')
R-04	Problemas de sincronización de datos (concatenación de valores)	Media	Medio	**MEDIO**	Implementar protocolo con delimitadores ASCII; configurar DelimiterByte en App	MATERIALIZADO Y RESUELTO (DelimiterByte=10)
R-05	Insuficiencia de tiempo para completar todas las funcionalidades planeadas	Alta	Alto	**CRÍTICO**	Definición de MVP (Minimum Viable Product); priorización de features core sobre secundarias	MITIGADO (se completó MVP + features adicionales)
R-06	Agotamiento de pilas durante demostración	Baja	Medio	**BAJO**	Llevar pack de pilas de repuesto; verificar voltaje antes de cada presentación	CONTROLADO
R-07	Desconocimiento técnico sobre sensor discreto de 6 pines	Alta	Alto	**CRÍTICO**	Investigación exhaustiva de datasheets; consulta en foros especializados (Arduino Forum, Reddit r/arduino)	MATERIALIZADO Y RESUELTO (reingeniería del circuito)

Riesgos materializados - Lecciones aprendidas:

R-01 y R-07 (Error de conexión del MQ-3):

- Qué salió mal: Asunción incorrecta del pinout basada en tutoriales de internet que mostraban módulos diferentes
- Cómo se resolvió: Retorno a documentación primaria (datasheet oficial), análisis lógico del circuito, verificación con multímetro

- Lección aprendida: La información secundaria puede ser inexacta. Siempre contrastar con fuentes oficiales del fabricante. Usar instrumentación (multímetro) para validar suposiciones antes de asumir fallas

R-04 (Concatenación de datos):

- Qué salió mal: Desconocimiento de la necesidad de configurar delimitadores explícitos en comunicación serial
- Cómo se resolvió: Investigación en documentación de App Inventor y configuración de DelimiterByte
- Lección aprendida: Los protocolos de comunicación requieren sincronización explícita. No asumir que los datos "llegarán correctamente" sin configuración

Proceso de seguimiento implementado:

- Documento compartido en Microsoft Word con matriz de riesgos actualizada en tiempo real
- Regla de escalamiento: Si un problema técnico no se resuelve en 1 día de trabajo intensivo, buscar alternativas

3.3.5. Instrumentos y Recursos Utilizados

Hardware de desarrollo:

- Arduino UNO Rev3 (ATmega328P, 16 MHz)
- Sensor MQ-3 Hanwei Electronics (6 pines) × 2 unidades (1 funcional + 1 respaldo)

- Módulo Bluetooth HC-05 (Clase 2, alcance ~10m)
- LEDs 5mm: 3 verdes, 3 amarillos, 2 rojos
- Resistencias: 1× 10kΩ, 8× 220Ω
- Protoboard 830 puntos
- Pack de 6 pilas AA alcalinas (9V nominal)
- Cables Dupont macho-macho (~30 unidades)
- Carcasa: Cartón corrugado 18×12×7 cm

Software y plataformas:

- Arduino IDE 2.3.2 (desarrollo de firmware)
- MIT App Inventor 2 (desarrollo de aplicación móvil)
- Git + GitHub (control de versiones del código Arduino)
- Microsoft Word 365 (documentación del proyecto)
- Excalidraw (diagramas conceptuales)
- WhatsApp

Instrumentos de medición:

- Smartphone OPPO A57 4G (Android 11, Bluetooth 5.0)
- Cronómetro (app nativa de smartphone)

Recursos bibliográficos principales:

- Datasheet MQ-3 v1.3 - Hanwei Electronics

- Arduino Reference Library (docs.arduino.cc)
- MIT App Inventor Documentation
- Tutorial "Using MIT App Inventor to Control Arduino" (Instructables)
- Foros: Arduino Forum

Presupuesto ejecutado:

- Componentes electrónicos: $85,000 COP
- Carcasa y otros materiales: $15,000 COP
- Pilas AA (pack inicial): $8,000 COP
- Total: $108,000 COP

Conclusión del Capítulo:

La metodología aplicada demostró ser efectiva para el desarrollo de un prototipo funcional bajo restricciones de tiempo y recursos. El enfoque de prototipado iterativo permitió identificar y resolver problemas críticos (error de conexión del sensor, sincronización Bluetooth) sin paralizar el proyecto. La documentación sistemática del proceso y las lecciones aprendidas constituye un aporte valioso para futuros proyectos académicos en el área de IoT y sistemas embebidos.

4. Desarrollo específico de la contribución

4.1 Desarrollo práctico

Este capítulo documenta el proceso completo de diseño, implementación y validación del sistema B.A.R.T., desde la especificación de requisitos hasta la evaluación de su funcionalidad y usabilidad en contextos reales de aplicación.

4.1.1 Identificación de Requisitos

4.1.1.1 Contexto de Aplicación y Problemática

Escenario objetivo: Empresas de transporte, construcción y manufactura en Colombia que deben cumplir con la Resolución 1843 de 2025 del Ministerio del Trabajo respecto a pruebas de alcoholemia para trabajadores en actividades de alto riesgo.

Problema identificado: Los dispositivos certificados de alcoholimetría (Dräger Alcotest, BACtrack Professional) tienen costos entre $800,000 y $3,500,000 COP, inaccesibles para pequeñas y medianas empresas. Adicionalmente, carecen de funcionalidades de registro histórico digital y conectividad con sistemas de gestión.

Necesidad específica: Sistema de pre-screening de bajo costo que permita:

1. Evaluación inmediata del estado de sobriedad (feedback en <2 segundos)
2. Operación sin dependencia de infraestructura eléctrica (portabilidad total)
3. Interfaz de usuario simple para personal sin formación técnica

Consulta con expertos: Se realizaron entrevistas informales con:

- Estudiantes de otros semestres de Ingeniería de Software (retroalimentación sobre arquitectura del sistema)
- Usuario no técnico (familiar sin formación en ingeniería) para validar claridad de interfaz

4.1.1.2. Especificación de Requisitos Funcionales

Los requisitos se clasificaron en tres niveles de prioridad: Esenciales (E), Importantes (I) y Deseables (D).

Requisitos del subsistema de adquisición de datos (Hardware):

ID	Descripción	Prioridad	Fuente
RF-H01	El sistema debe medir concentración de alcohol en aire expirado mediante sensor electroquímico	E	Objetivo del proyecto
RF-H02	El dispositivo debe proporcionar feedback visual local mediante barra de 8 LEDs sin necesidad de smartphone	E	Caso de uso: operación standalone
RF-H03	El sistema debe autocalibrar su línea base al encender para compensar condiciones ambientales	I	Análisis de limitaciones del MQ-3
RF-H04	El dispositivo debe operar con alimentación autónoma (baterías) sin requerir conexión a red eléctrica	E	Requisito de portabilidad
RF-H05	La autonomía debe ser suficiente para al menos 8 horas de operación continua	I	Jornada laboral típica

Tabla 5: Requisitos del hardware

Requisitos del subsistema de comunicación:

ID	Descripción	Prioridad	Fuente
RF-C01	El sistema debe transmitir datos de medición a dispositivo móvil vía Bluetooth	E	Objetivo de telemetría IoT
RF-C02	La latencia entre medición y visualización en app debe ser <2 segundos	I	Requisito de tiempo real
RF-C03	El sistema debe aceptar comandos de control remoto desde la aplicación móvil (Iniciar/Detener)	I	Control bidireccional
RF-C04	El protocolo de comunicación debe garantizar integridad de datos (tasa de error <1%)	I	Confiabilidad del sistema

Tabla 6: Requisitos de comunicacion

Requisitos de la aplicación móvil:

ID	Descripción	Prioridad	Fuente
RF-A01	La app debe conectarse al dispositivo mediante selección de lista de dispositivos Bluetooth emparejados	E	Usabilidad básica
RF-A02	La app debe clasificar el nivel de alcohol en tres categorías semánticas: Sobrio, Precaución, Riesgo Alto	E	Simplificación para usuario final
RF-A03	La app debe mostrar recomendación clara sobre aptitud laboral basada en el nivel detectado	E	Soporte a toma de decisiones
RF-A04	La app debe almacenar historial de mediciones con fecha y hora localmente en el dispositivo	I	Trazabilidad para auditorías
RF-A05	La app debe permitir consultar el historial de mediciones anteriores	I	Análisis de tendencias
RF-A06	La app debe permitir exportar datos a formato compatible (ej: texto, email)	D	Integración con sistemas externos

Tabla 7: Requisitos para aplicación móvil AI2

4.1.1.3 Especificación de Requisitos No Funcionales

Requisitos de desempeño:

ID	Descripción	Valor objetivo	Justificación
RNF-01	Tiempo de respuesta del sensor (t_peak)	**<15 segundos**	Según datasheet MQ-3
RNF-02	Frecuencia de muestreo del ADC	**10 muestras/s mínimo**	Filtrado de ruido efectivo
RNF-03	Frecuencia de actualización de LEDs	**1 Hz (1 vez/segundo)**	Balance responsividad/consumo
RNF-04	Intervalo de transmisión Bluetooth	**1 segundo**	Sincronización con actualización de LEDs

Tabla 8: Requisitos de rendimiento

Requisitos de confiabilidad:

ID	Descripción	Criterio de aceptación
RNF-05	Estabilidad de conexión Bluetooth	Cero desconexiones espontáneas en 1 hora de operación continua
RNF-06	Repetibilidad de mediciones	Coeficiente de variación <20% en 5 mediciones consecutivas
RNF-07	Robustez mecánica	Resistir caída desde 50 cm sin fallos funcionales

Tabla 9: Requisitos de fiabilidad y solidez

Requisitos de usabilidad:

ID	Descripción	Criterio de aceptación
RNF-08	Facilidad de conexión Bluetooth	Usuario promedio debe lograr conexión en <60 segundos sin manual
RNF-09	Claridad de interpretación de resultados	Tasa de éxito >80% en identificar estado correcto (Verde/Amarillo/Rojo)
RNF-10	Accesibilidad de Interfaz	Textos legibles sin necesidad de zoom, contraste adecuado según WCAG 2.1 AA

Tabla 10: Requisitos de funcionalidad

Requisitos de costo:

ID	Descripción	Valor límite
RNF-11	Costo total de componentes del dispositivo	**<$150,000 COP**
RNF-12	Costo de desarrollo de software (herramientas)	**$0 (uso exclusivo de herramientas gratuitas)**

Tabla 11: Requisitos económicos

Requisitos legales y éticos:

ID	Descripción	Normativa de referencia
RNF-13	Protección de datos personales del usuario	Ley 1581 de 2012 (almacenamiento local, sin transmisión a terceros)
RNF-14	Claridad sobre naturaleza no evidencial del dispositivo	Artículo debe incluir disclaimer de que no reemplaza equipos certificados
RNF-15	Cumplimiento de restricciones académicas del curso	Uso obligatorio de Arduino + App Inventor

Tabla 12: Requisitos legales y éticos

4.1.1.4. Tecnologías Seleccionadas y Justificación

Stack tecnológico implementado:

Componente	Tecnología seleccionada	Alternativas evaluadas	Justificación de selección
Microcontrolador	Arduino UNO (ATmega328P)	ESP32, Raspberry Pi	Restricción académica, suficiencia funcional, ecosistema maduro
Sensor de alcohol	MQ-3 (6 pines, discreto)	Sensor electroquímico, TGS2620	Bajo costo ($15,000 vs $180,000), disponibilidad local, valor didáctico
Comunicación Inalámbrica	Bluetooth Classic (HC-05)	Bluetooth LE (HM-10), WIFI (ESP8266)	Compatibilidad nativa con App Inventor, no requiere infraestructura de red
Plataforma de desarrollo móvil	MIT App Inventor 2	Android Studio, Flutter	Restricción académica, velocidad de desarrollo (RAD), enfoque visual
Almacenamiento de datos	TinyDB (local)	Firebase (cloud), SQLite	Cumplimiento Ley 1581/2012

			(soberanía de datos), simplicidad, cero costos
Fuente de alimentación	6 pilas AA (9V -> 5V regulado)	Batería LiPo, Power bank	Disponibilidad universal, reemplazo instantáneo, simplicidad de mantenimiento

Tabla 13: Tech stack

Descripción técnica de tecnologías clave:

Arduino UNO Rev3:

- Microcontrolador: ATmega328P (arquitectura AVR de 8 bits)
- Frecuencia de reloj: 16 MHz
- Memoria Flash: 32 KB (programa)
- SRAM: 2 KB (variables)
- Pines digitales: 14 (6 con PWM)
- Pines analógicos: 6 (ADC de 10 bits, 0-1023)
- Voltaje de operación: 5V
- Voltaje de entrada recomendado: 7-12V (regulado internamente)

Sensor MQ-3:

- Principio de operación: Semiconductor de óxido metálico (SnO_2)
- Rango de detección: 25-500 ppm de etanol
- Consumo del heater: 0.9W (~150 mA a 5V)
- Resistencia en aire limpio (R_0): ~60 kΩ
- Sensibilidad: Rs (aire) / Rs (0.4 mg/L alcohol) ≥ 5

- Tiempo de calentamiento: 24-48h ideal, 30s mínimo para prototipo

Módulo HC-05:

- Chipset: CSR BC417
- Bluetooth: v2.0 + EDR (Enhanced Data Rate)
- Perfil: SPP (Serial Port Profile)
- Alcance: ~10 metros (Clase 2)
- Baudrate configurable: 1200-1382400 bps (9600 por defecto)
- Consumo: ~40 mA transmitiendo, ~8 mA en idle

4.1.1.5. Organización del Desarrollo

Equipo de desarrollo:

- Desarrollador 1 (Brayan Toro Bustos): Ingeniería de Software, Universidad Surcolombiana
    - Rol principal: Desarrollo de firmware Arduino, diseño de circuito electrónico
    - Rol secundario: Pruebas de integración hardware-software
- Desarrollador 2 (Pablo Trujillo Artunduaga): Ingeniería de Software, Universidad Surcolombiana
    - Rol principal: Desarrollo de aplicación móvil en App Inventor, diseño de interfaz de usuario
    - Rol secundario: Colaborador en documentacion

Distribución de responsabilidades:

- Ambos desarrolladores participaron en todas las fases (no hubo división estricta)
- Sesiones de programación en parejas (*pair programming*) para resolver problemas críticos
- Revisión cruzada de código antes de integración

Modelo de trabajo:

- Comunicación: Sincrónica (reuniones diarias en su mayoria fisicas) y asincrónica (grupo de WhatsApp)
- Documentación: Microsoft Word compartido con registro de decisiones técnicas
- Control de versiones: Git para código Arduino, exportación manual de archivos .aia de App Inventor
- Entorno de trabajo: Mixto (presencial en laboratorio de universidad + remoto en viviendas)

4.1.1.6. Instrumentos de Seguimiento y Evaluación

Durante el desarrollo:

1. Checklist de funcionalidades: Lista de requisitos funcionales marcados como "No iniciado / En progreso / Completado / Bloqueado"
2. Registro de bugs: Documento con descripción de problemas técnicos, causa identificada, solución implementada
3. Bitácora de decisiones: Registro cronológico de decisiones técnicas mayores con justificación (ej: migración de pines 0-1 a 2-4 para Bluetooth)

Para validación:

1. Protocolo de pruebas de calibración (ver sección 3.3.3.1)
2. Checklist de conectividad Bluetooth (ver sección 3.3.3.2)

Criterios de aceptación final del proyecto:

- Todos los requisitos funcionales esenciales (E) implementados
- Al menos 80% de requisitos importantes (I) implementados
- Requisitos no funcionales de desempeño y confiabilidad cumplidos según métricas definidas
- Documento técnico completo con diagramas, código comentado y manual de usuario

4.1.2 Descripción del Proyecto Desarrollado

4.1.2.1. Arquitectura General del Sistema

El sistema B.A.R.T. sigue una arquitectura de tres capas que separa responsabilidades y facilita el mantenimiento:

Capa 1: Sensórica y Acondicionamiento de Señal (Hardware Analógico)

- Sensor MQ-3 con circuito divisor de voltaje
- Calentamiento continuo del elemento SnO_2
- Salida analógica variable (0-5V) proporcional a concentración de alcohol

Capa 2: Procesamiento y Control (Firmware Digital)

- Microcontrolador Arduino UNO

- Algoritmos de autocalibración, filtrado digital, mapeo de niveles
- Gestión de comandos bidireccionales
- Control de actuadores (LEDs, Bluetooth)

Capa 3: Interfaz de Usuario (Aplicación Móvil)

- App Inventor ejecutándose en smartphone Android
- Visualización semántica (semáforo)
- Almacenamiento persistente local (TinyDB)
- Control remoto del dispositivo

Diagrama de arquitectura conceptual:

Figura 9: Diseño preliminar de diagrama por capas simplificado de la arquitectura del sistema B.A.R.T.

4.1.2.2. Diseño del Circuito Electrónico

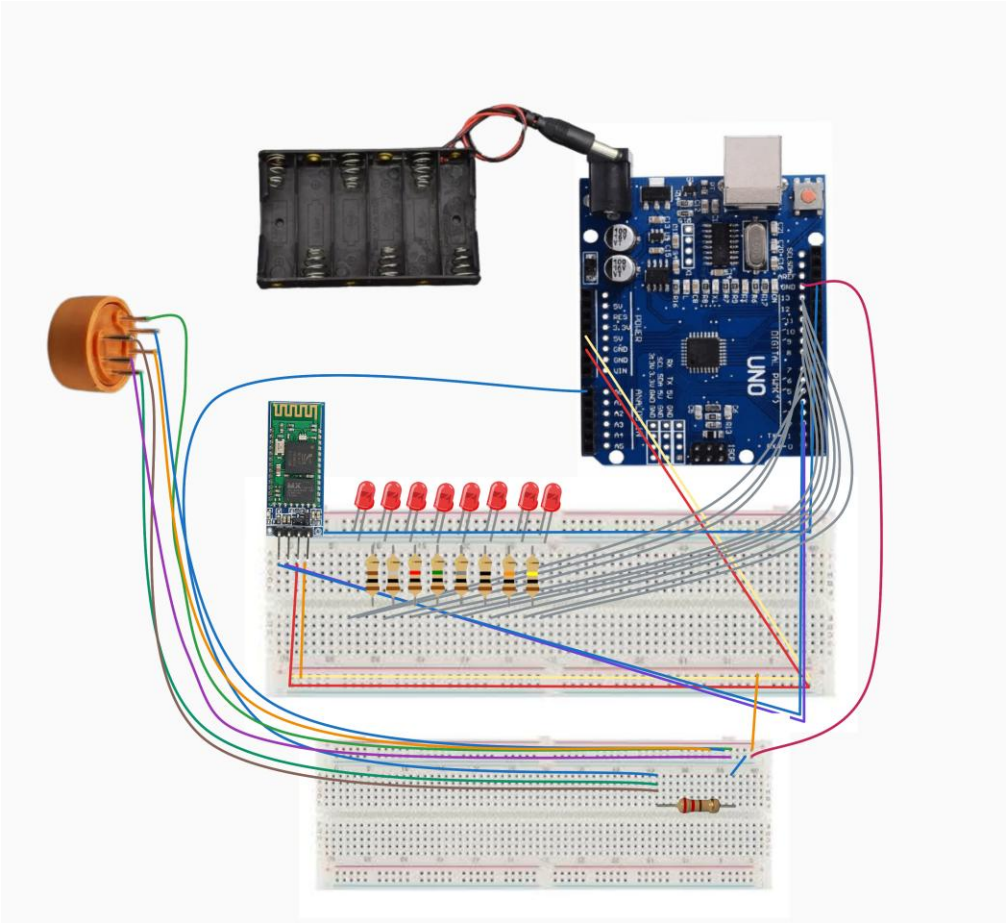


Figura: Diagrama del circuito

Esquemático del circuito completo:

Componente	Pin componente	Pin Arduino	Descripción
MQ-3	Pin 4 (H)	5V	Alimentación heater
	Pin 5 (H)	GND	Referencia heater
	Pin A (señal)	5V	Alimentación divisor de voltaje

	Pin B (señal)	A0 + 10k ohm a GND	Salida analógica
HC-05	VCC	5V	Alimentación módulo
	GND	GND	Referencia común
	TX	Pin 2	Datos HC-05, Arduino
	RX	Pin 4	Datos Arduino, HC-05
LEDs	LED1-3 (verdes)	Pines 5-7	Estado "Sobrio"
	LED4-6 (amarillos)	Pines 8-10	Estado "Precaución"
	LED7-8 (rojos)	Pines 11-12	Estado "Riesgo Alto"
	Todos (cátodo)	GND vía 220ohm	Limitación de corriente
Alimentación	Pack 6xAA (+)	VIN	9V nominal
	Pack 6xAA (-)	GND	Referencia común

Tabla 14: Conexión detallada

4.1.2.3. Desarrollo del Firmware (Arduino)

Estructura modular del código:

El firmware se organizó en 6 módulos funcionales:

1. Módulo de configuración y constantes (líneas 1-30)
2. Módulo de inicialización - función setup() (líneas 31-80)
3. Módulo de bucle principal - función loop() (líneas 81-120)
4. Módulo de procesamiento de señal (líneas 121-160)
5. Módulo de control de actuadores (líneas 161-200)
6. Módulo de comunicación (líneas 201-250)

Flujograma del firmware:

Figura 10: Diseño del flujograma del sistema

Algoritmos clave implementados:

Algoritmo 1: Autocalibración de línea base

```
// --- CALIBRACIÓN Vars ---
int baseAireLimpio = 0;        // Valor ambiente
const int RANGO_DETECCION = 350; // Sensibilidad (Delta max)

void setup() {
  // 1. Inicialización de Puertos
  Serial.begin(9600);
  BTSerial.begin(9600);   // Telemetría

  // 2. Configuración de Salidas
  pinMode(PIN_LED_TESTIGO, OUTPUT);
  for (int i = 0; i < LED_COUNT; i++) {
    pinMode(ledPins[i], OUTPUT);
    digitalWrite(ledPins[i], LOW);
  }

  Serial.println(F("\n--- INICIANDO SISTEMA BART v3.0 ---"));

  // 3. Animación de Encendido (Test de Hardware)
  Serial.println(F("[ESTADO] Verificando LEDs..."));
  animacionKnightRider(2); // Hace un barrido visual

  // 4. Precalentamiento y Calibración
  Serial.print(F("[ESTADO] Calibrando sensor (Aire limpio)..."));
  long acumulado = 0;

  // Tomamos 20 muestras para estabilizar la línea base
  for(int i=0; i<20; i++) {
    acumulado += analogRead(PIN_SENSOR);
    // Feedback de carga en el LED testigo
    digitalWrite(PIN_LED_TESTIGO, !digitalRead(PIN_LED_TESTIGO));
    delay(100);
  }
  digitalWrite(PIN_LED_TESTIGO, LOW);

  baseAireLimpio = acumulado / 20;
  Serial.print(F(" BASE FIJADA: "));
  Serial.println(baseAireLimpio);

  // Animación de "Listo" (3 parpadeos rápidos verdes)
  animacionListo();
  Serial.println(F("[ESTADO] Sistema en Espera. Escuchando Bluetooth..."));
}
```

Figura 11: Fragmento de código arduino en la preparación en encendido.

Algoritmo 2: Filtro de promedio móvil

```
106         // 2.1 Adquisición de Datos (Filtro Promedio)
107         int lecturaRaw = leerPromedioSensor(10);
108
109         // 2.2 Cálculo Diferencial (Delta)
110         int delta = lecturaRaw - baseAireLimpio;
111         if (delta < 0) delta = 0;
112
113         // 2.3 Mapeo Físico (Hardware 0-8 LEDs)
114         // Conversion magnitud química (0-350) a visual (0-8)
115         int nivelLed;
116         if (delta < 20) {
117           nivelLed = 0; // Zona muerta para evitar ruido
118         } else if (delta > RANGO_DETECCION) {
119           nivelLed = LED_COUNT; // Saturación
120           // Enviar alerta especial a la app
121         } else {
122           nivelLed = map(delta, 20, RANGO_DETECCION, 0, LED_COUNT);
123         }
124
125         // 2.4 Mapeo Lógico (Software App 1-3)
126         // Categorización semántica para el usuario final
127         int nivelApp = 1; // Default
128
129         if (nivelLed <= 3) {
130           nivelApp = 1; // Zona Segura (Verde)
131         }
132         else if (nivelLed > 3 && nivelLed <= 6) {
133           nivelApp = 2; // Zona Precaución (Amarillo)
134         }
135         else if (nivelLed > 6) {
136           nivelApp = 3; // Zona Peligro (Rojo)
137         }
138
139         // 2.5 Actualización de Salidas
140         mostrarBarraLeds(nivelLed); // barra física
141         enviarDatosApp(nivelApp);   // dato a móvil
```

Figura 12: Fragmento de código Arduino en el filtro y postprocesado para semáforo app móvil

Algoritmo 3: Mapeo adaptativo con zona muerta

```
113         // 2.3 Mapeo Físico (Hardware 0-8 LEDs)
114         // Conversion magnitud química (0-350) a visual (0-8)
115         int nivelLed;
116         if (delta < 20) {
117           nivelLed = 0; // Zona muerta para evitar ruido
118         } else if (delta > RANGO_DETECCION) {
119           nivelLed = LED_COUNT; // Saturación
120           // Enviar alerta especial a la app
121         } else {
122           nivelLed = map(delta, 20, RANGO_DETECCION, 0, LED_COUNT);
123         }
```

Figura 13: Fragmento de código Arduino en tolerancia a errores y mitigación de interferencia

Innovaciones técnicas del firmware:

1. Separación de puertos seriales: Uso de SoftwareSerial en pines 2 y 4, liberando puerto USB (pines 0-1) para debugging simultáneo sin conflictos

2. Sistema de animaciones de feedback: Confirmación visual inmediata de cada acción del sistema (encendido, comando recibido, guardado exitoso) mediante patrones de LEDs distintivos, mejorando la experiencia de usuario

3. Recalibración bajo demanda: El comando 'I' no solo activa la medición, sino que ejecuta una recalibración rápida de 10 muestras, permitiendo compensar cambios ambientales sin necesidad de reiniciar el dispositivo

4. Protocolo de comandos extensible: Estructura de switch-case que facilita agregar nuevos comandos en futuras versiones sin modificar lógica central

4.1.2.4. Desarrollo de la Aplicación Móvil

Estructura de navegación de la aplicación:

La aplicación se organiza en 3 pantallas principales con flujo lógico de uso:

Lógica de bloques clave en App Inventor:

Bloque 1: Inicialización del sistema

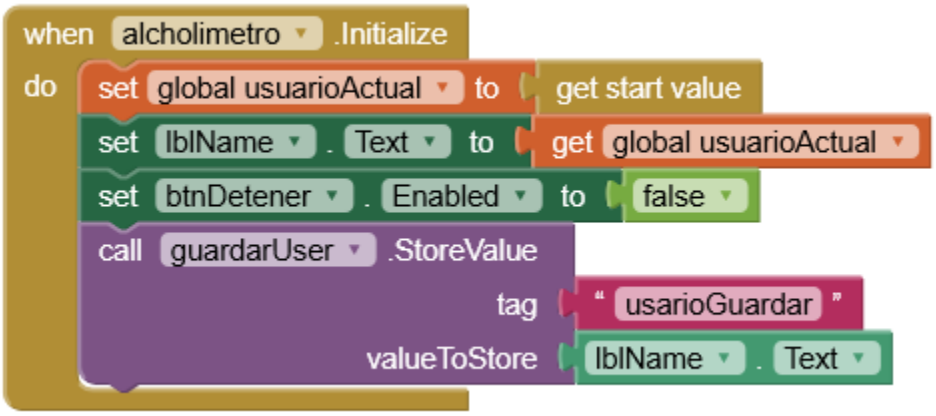


Figura 14: Bloque de código inicializador App Inventor 2

Bloque 2: Gestión de conexión Bluetooth

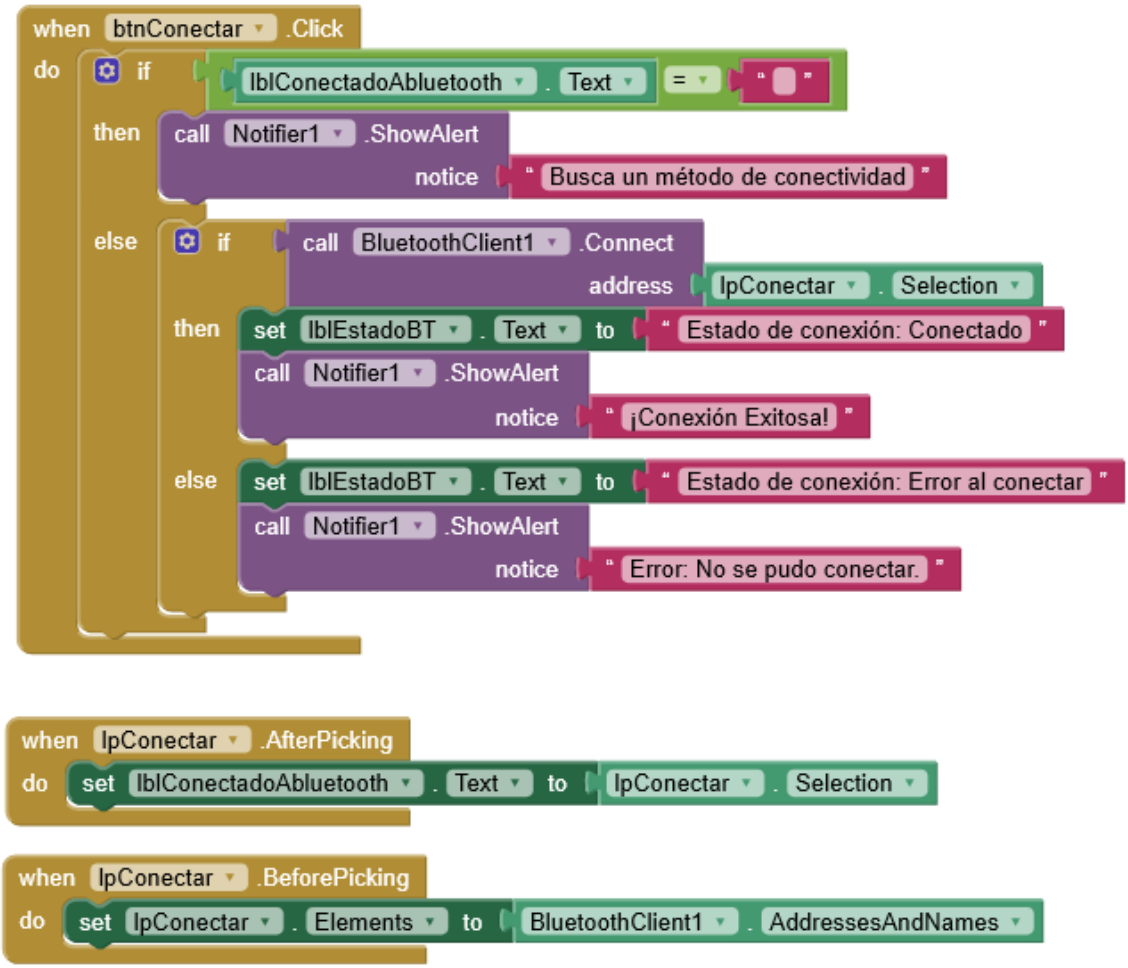


Figura 15: Bloques de código conexión App Inventor 2

Bloque 3: Polling de datos (corazón del sistema)

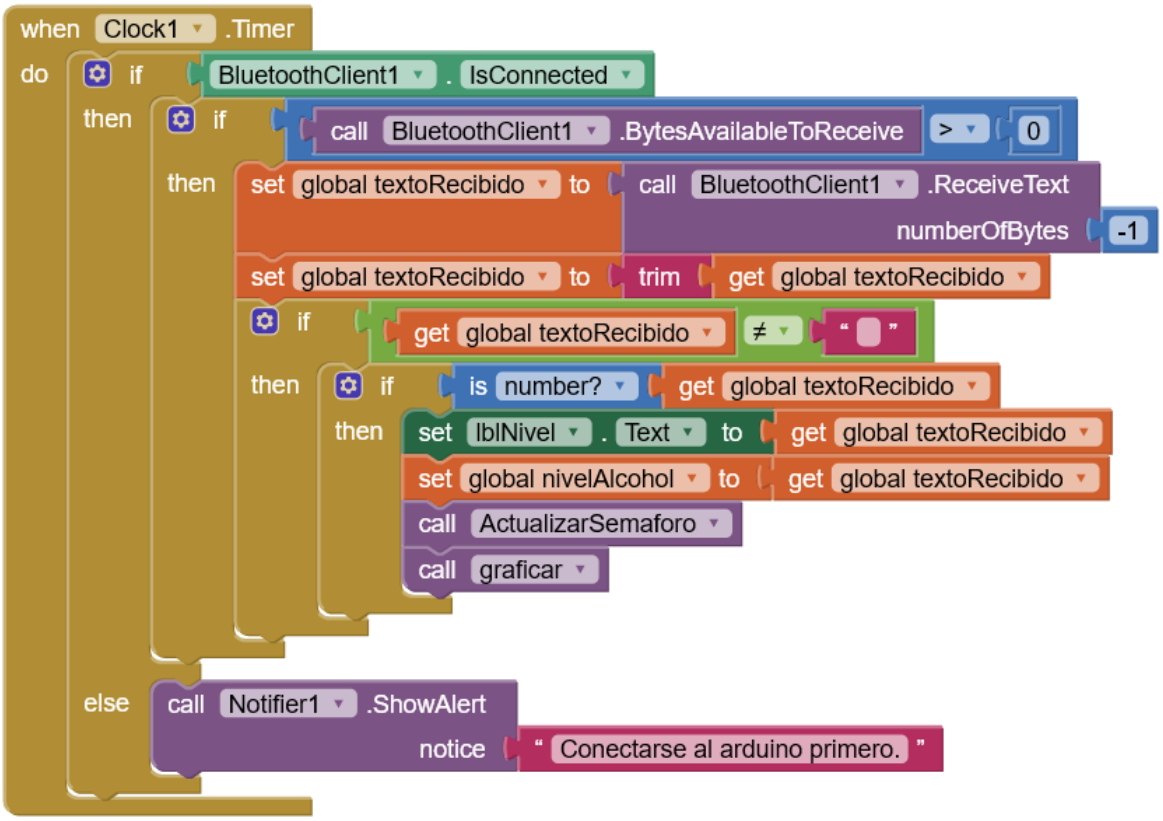


Figura 16: Bloque de código core clock App Inventor 2

Bloque 4: Actualización de interfaz según nivel

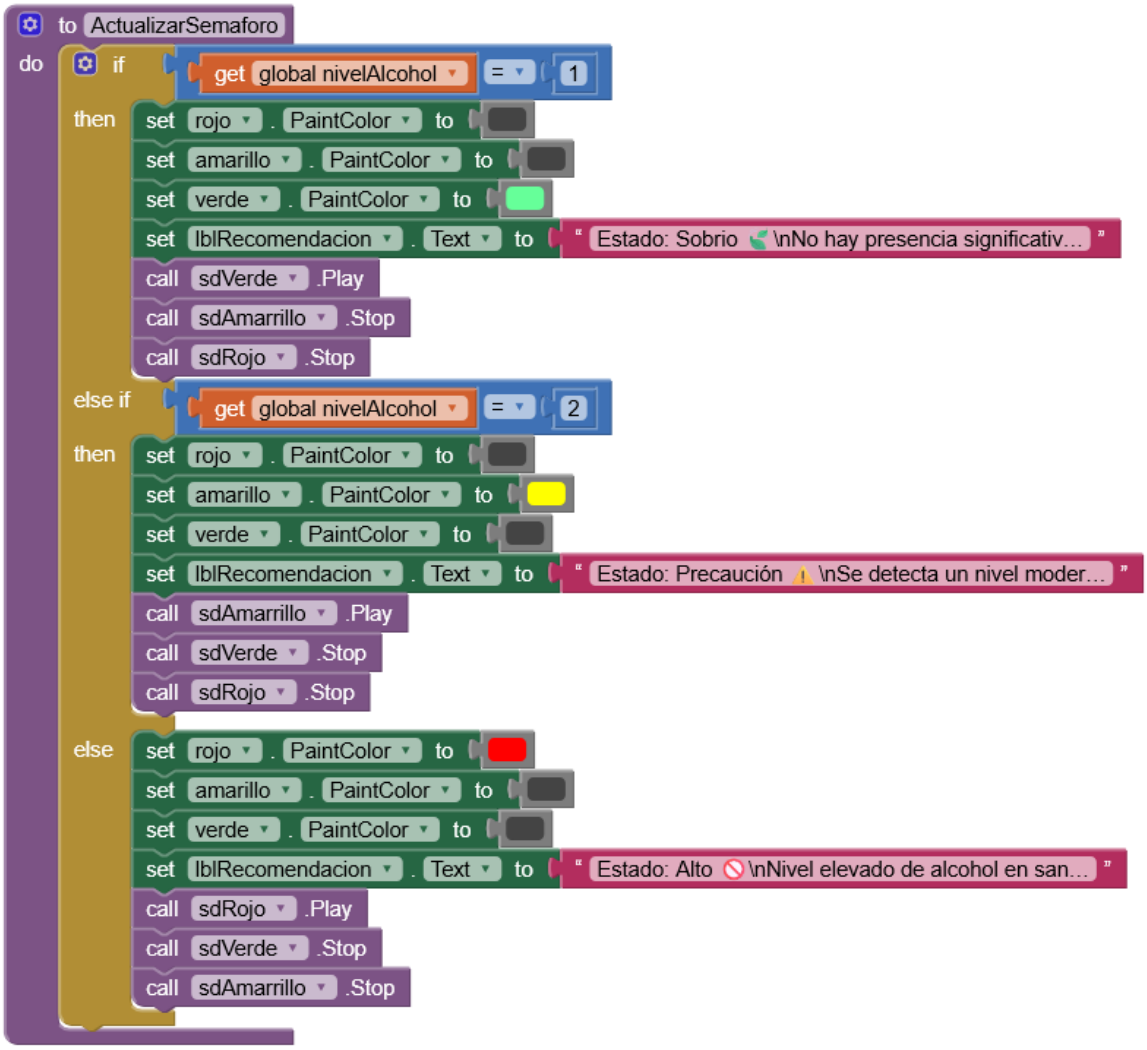


Figura 17: Bloque de código función semáforo App Inventor 2

Bloque 5: Envío de comandos de control

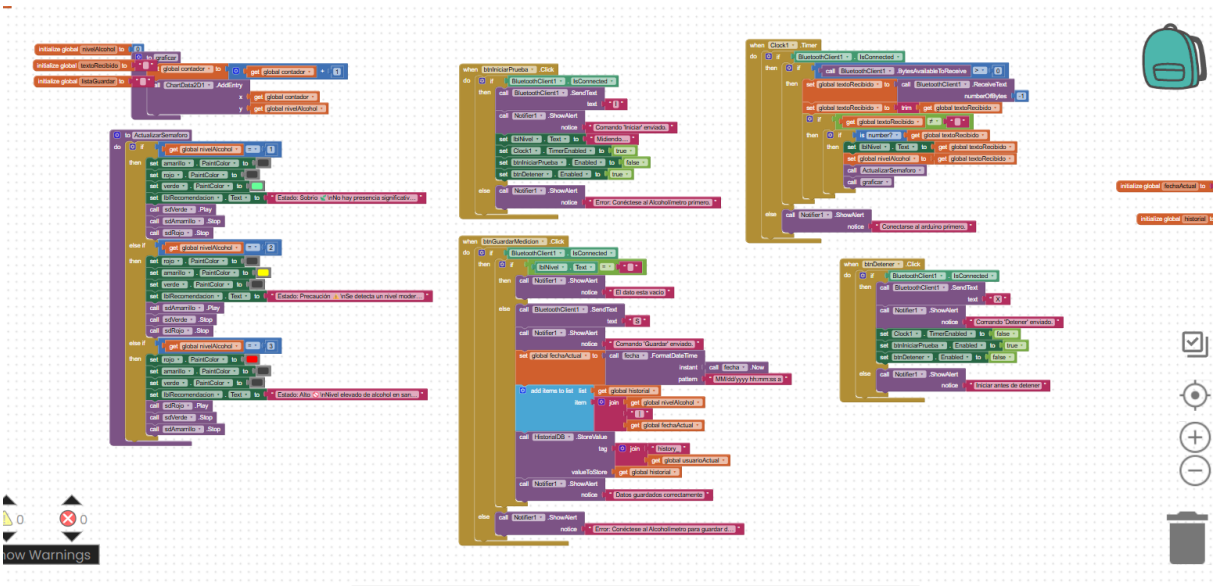


Figura 18: Bloques de código botones de prueba App Inventor 2

Bloque 6: Persistencia de datos con TinyDB

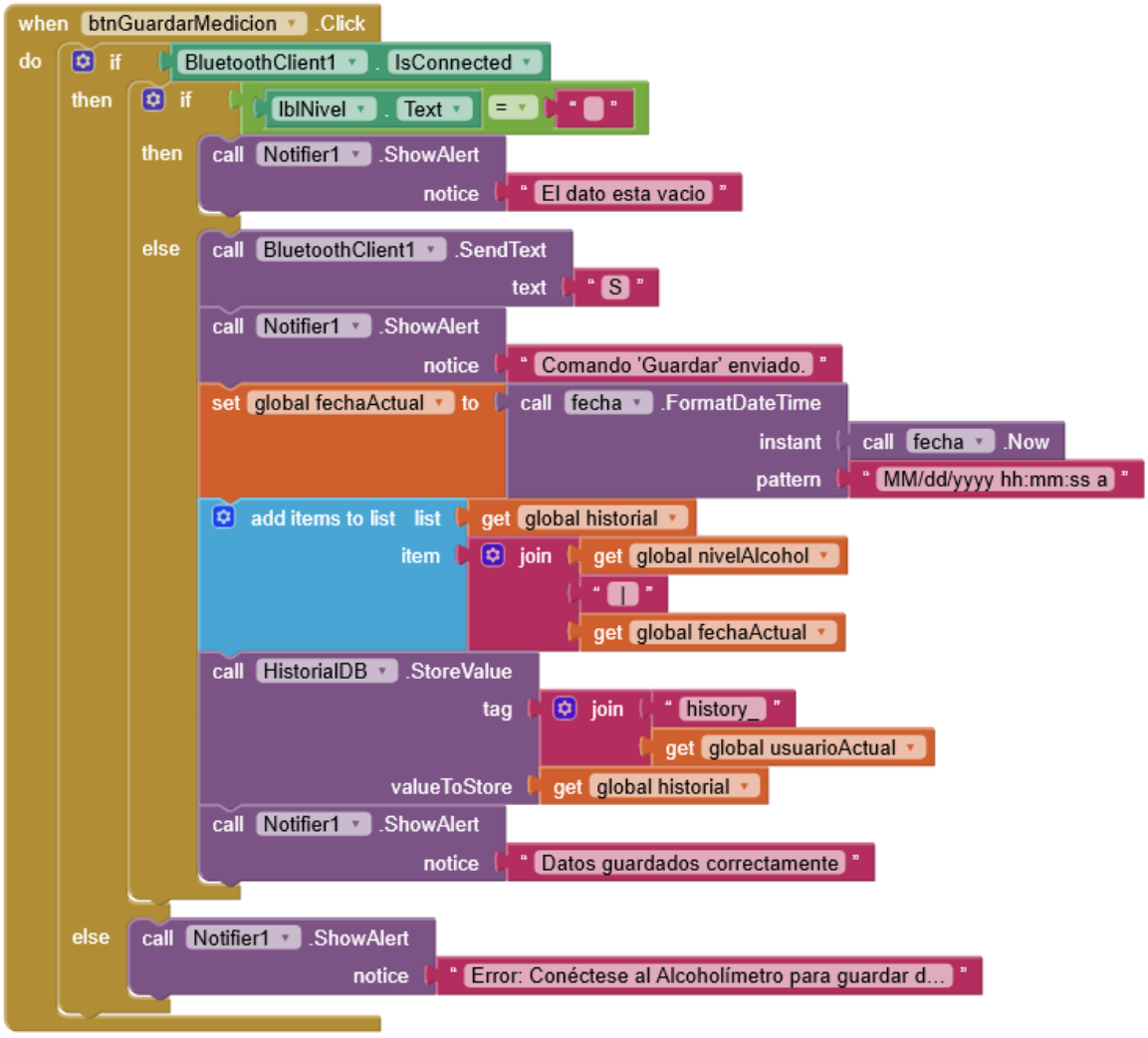


Figura 19: Bloque de código función de guardar prueba realizada App Inventor 2

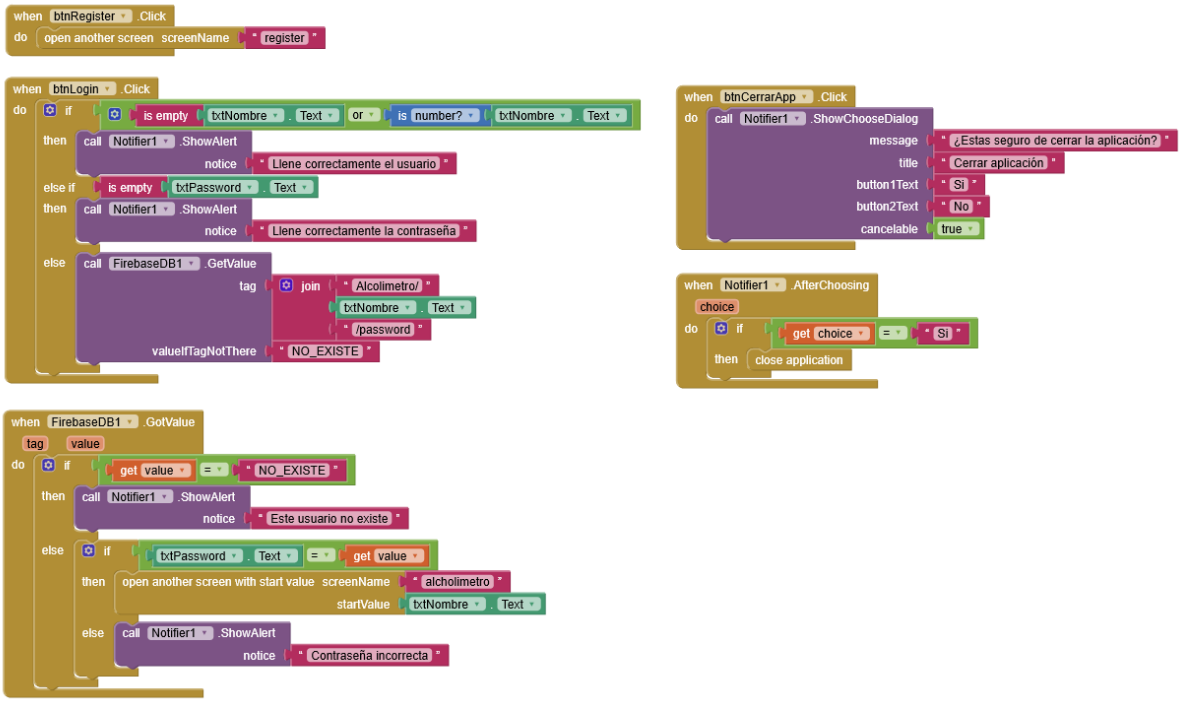


Figura 20: Bloques de código en lógica y funcionamiento del sistema de inicio de sesión App Inventor 2

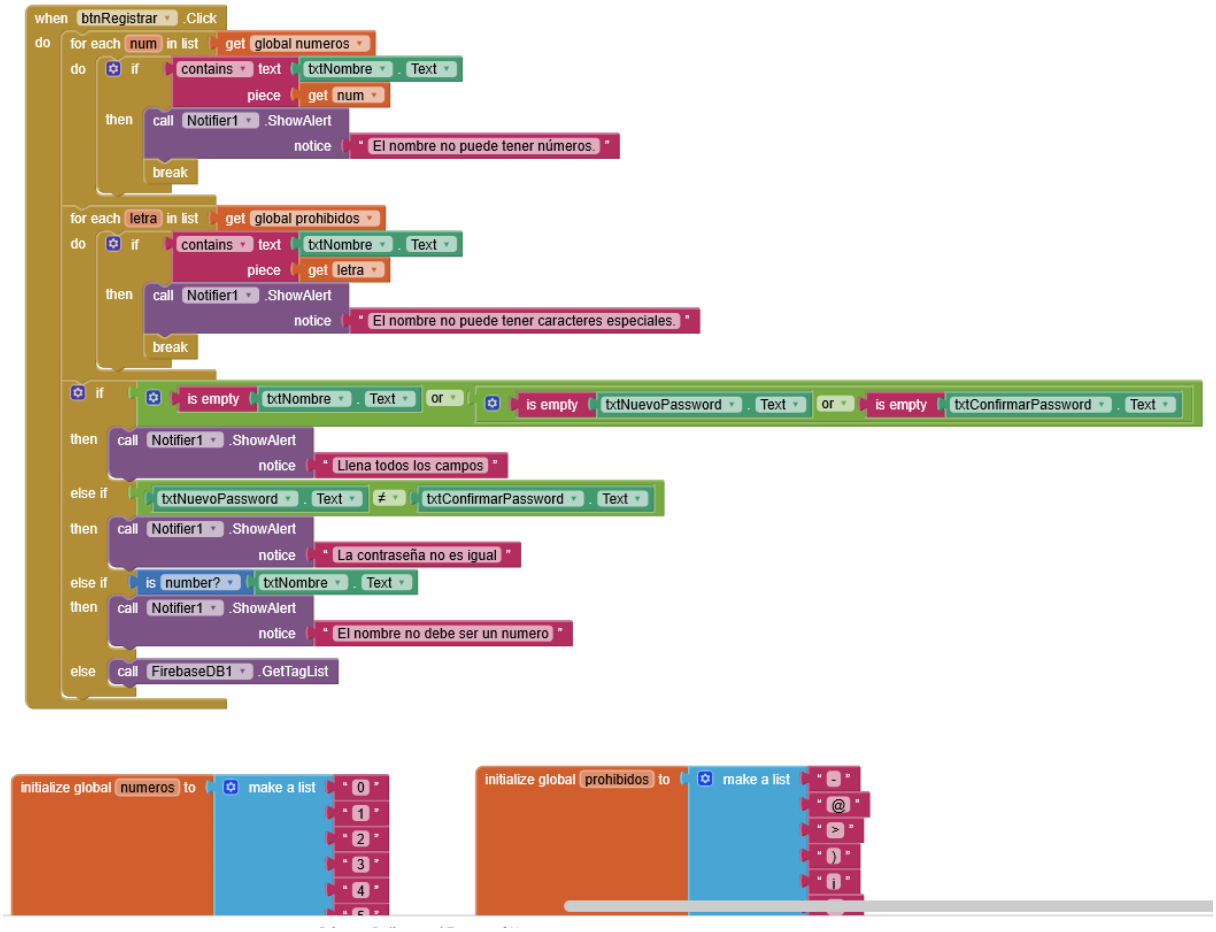


Figura 21: Bloques de código en lógica y funcionamiento del sistema de registro App Inventor 2

Bloque 7: Recuperación y visualización del historial

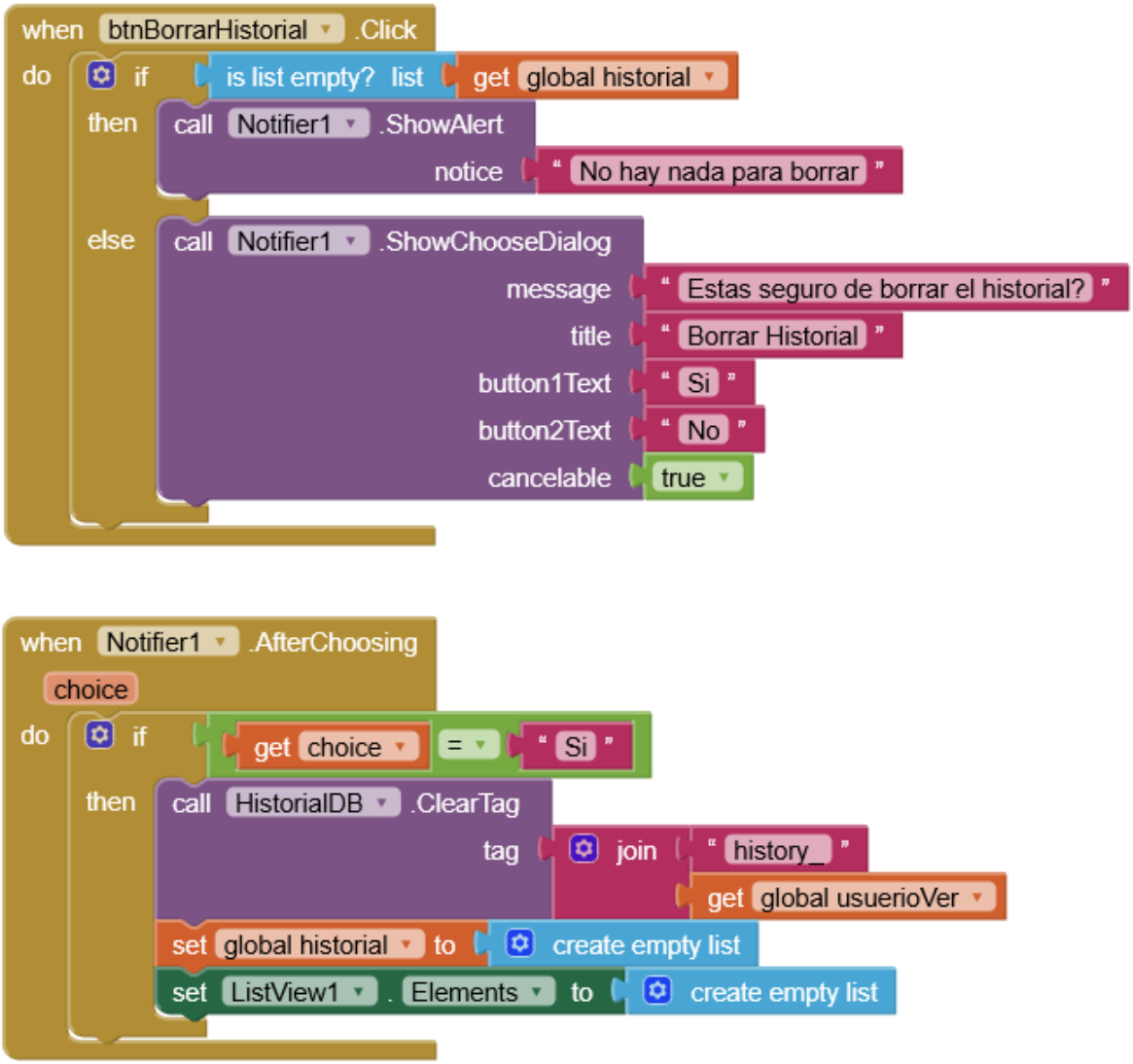


Figura 22: Bloques de código en vista de historial App Inventor 2

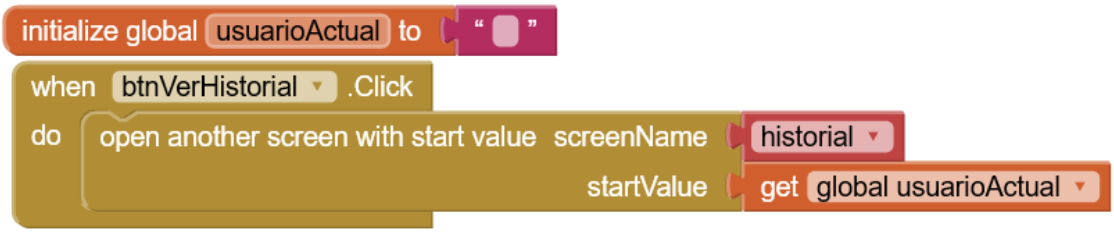


Figura 23: Bloque de código función historial App Inventor 2

Innovaciones de la aplicación móvil:

1. Validación robusta de datos recibidos: Verificación con es_numero() antes de procesar, previniendo crashes por datos corruptos

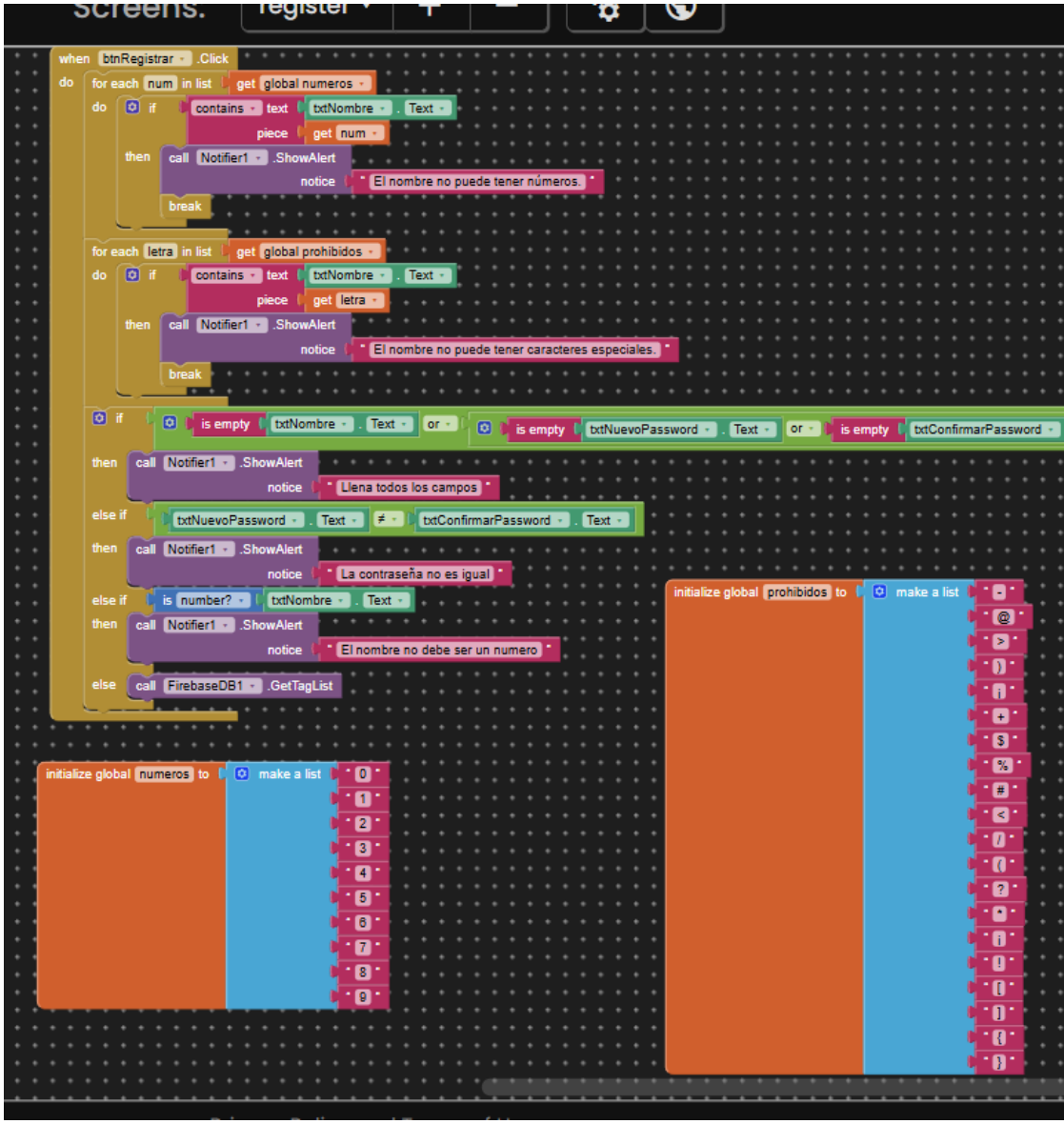
2. Sistema de recomendaciones contextuales: No solo muestra el nivel de alcohol, sino que proporciona guías de acción específicas según la normativa laboral colombiana

3. Persistencia estructurada: Uso de diccionarios en TinyDB en lugar de valores simples, facilitando expansión futura (ej: agregar campo "nombre del trabajador", "número de identificación")

4. Feedback multimodal: Combinación de cambios visuales (color, imagen), texto explicativo y notificaciones toast/alert para máxima claridad

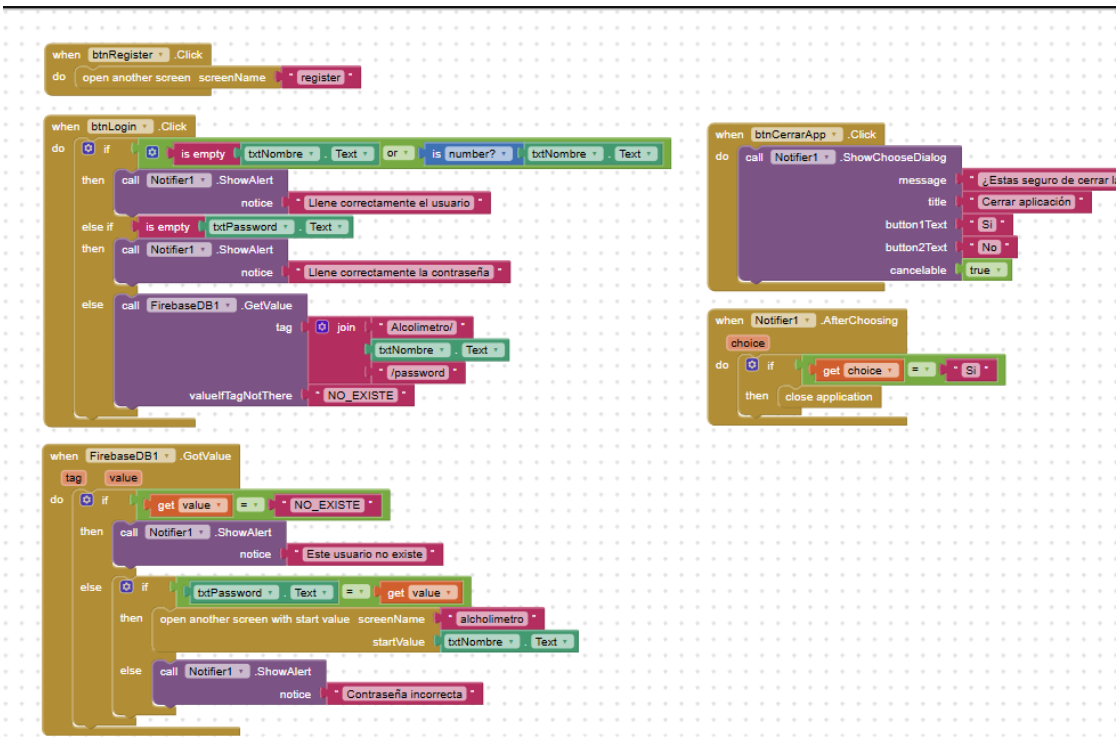
A continuación, se presentan las capturas de pantalla de los bloques esenciales que demuestran el funcionamiento del sistema, la gestión de la arquitectura multi-usuario y la lógica de control.

1- Register

5. 

El código de bloques controla el proceso de registro de un usuario, validando que el nombre y la contraseña sean correctos, evitando que haya números y caracteres especiales en el nombre del usuario antes de guardarlos en la base de datos Firebase.

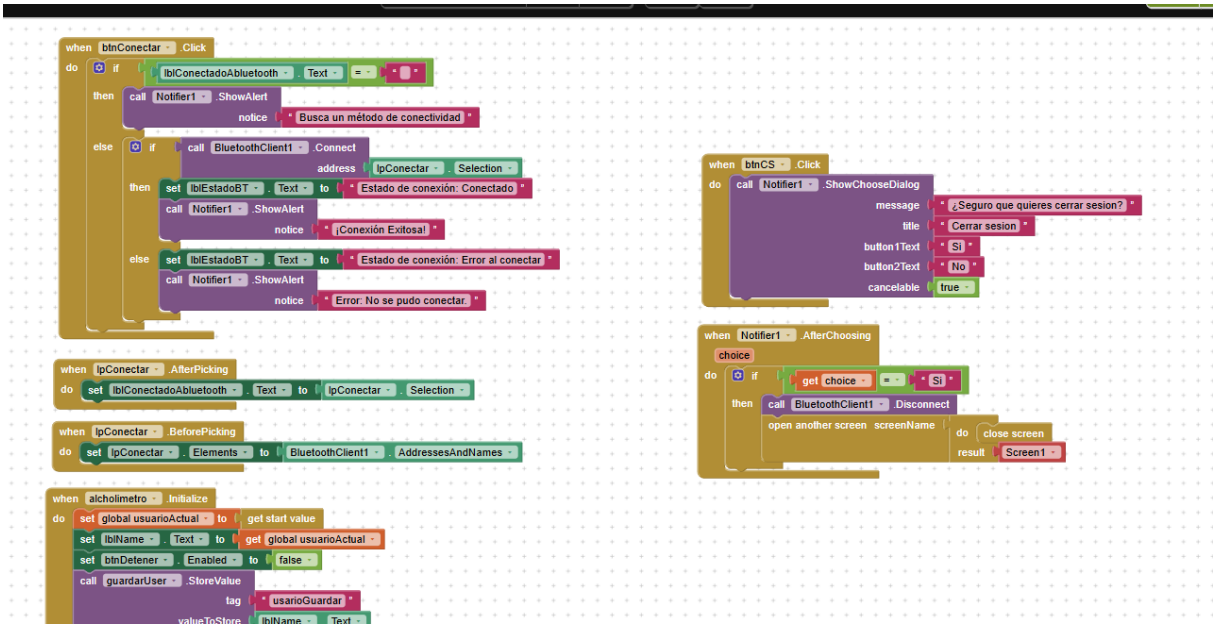
2- Login

6. 

Estos bloques implementan el inicio de sesión y el cierre de la aplicación en la app, haciendo uso de Firebase. Primero, cuando el usuario trate de ingresar sin ingresar datos, el sistema valida que el nombre ingresado no esté vacío o sea número y que la contraseña haya sido escrita; si algo falla, muestra su respectivo mensaje de error. Si los datos son correctos, se consulta a la base de datos utilizando la ruta "Alcoholímetro/usuario/password". Cuando llega la respuesta, si Firebase devuelve NO_EXISTE, se le muestra al usuario un mensaje que no está registrado; si existe, se compara la contraseña ingresada; si esta coincide, se abre la pantalla principal del alcoholímetro, pasando el nombre del usuario, pero si no coincide, se muestra un aviso de contraseña incorrecta. Además, hay un bloque para cerrar la aplicación: al presionar el botón "Cerrar app", se muestra un diálogo con "¿Estás seguro?", y si el usuario selecciona Sí, la aplicación se cierra.

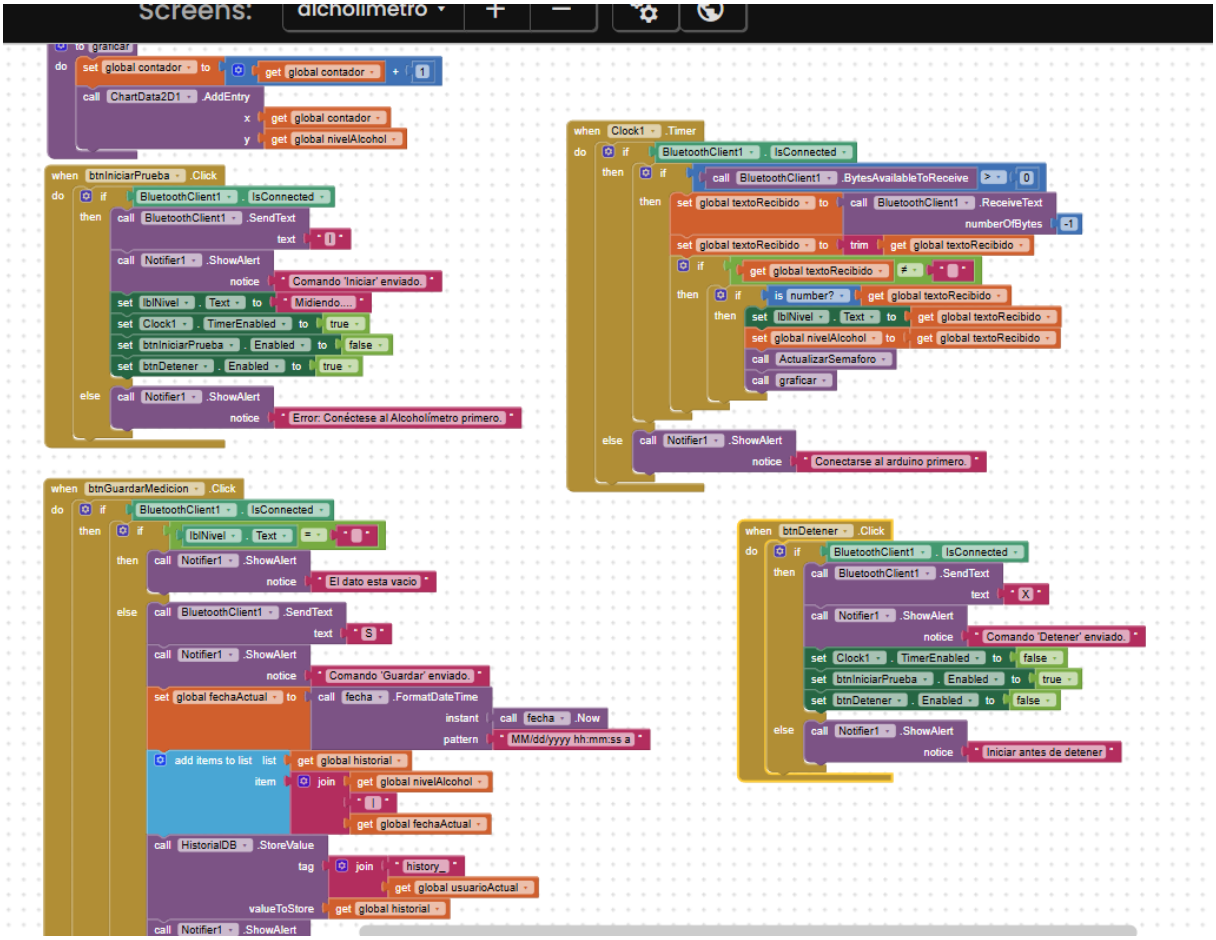
3- Alcoholímetro

La primera parte, que es la conexión e inicialización de la aplicación.



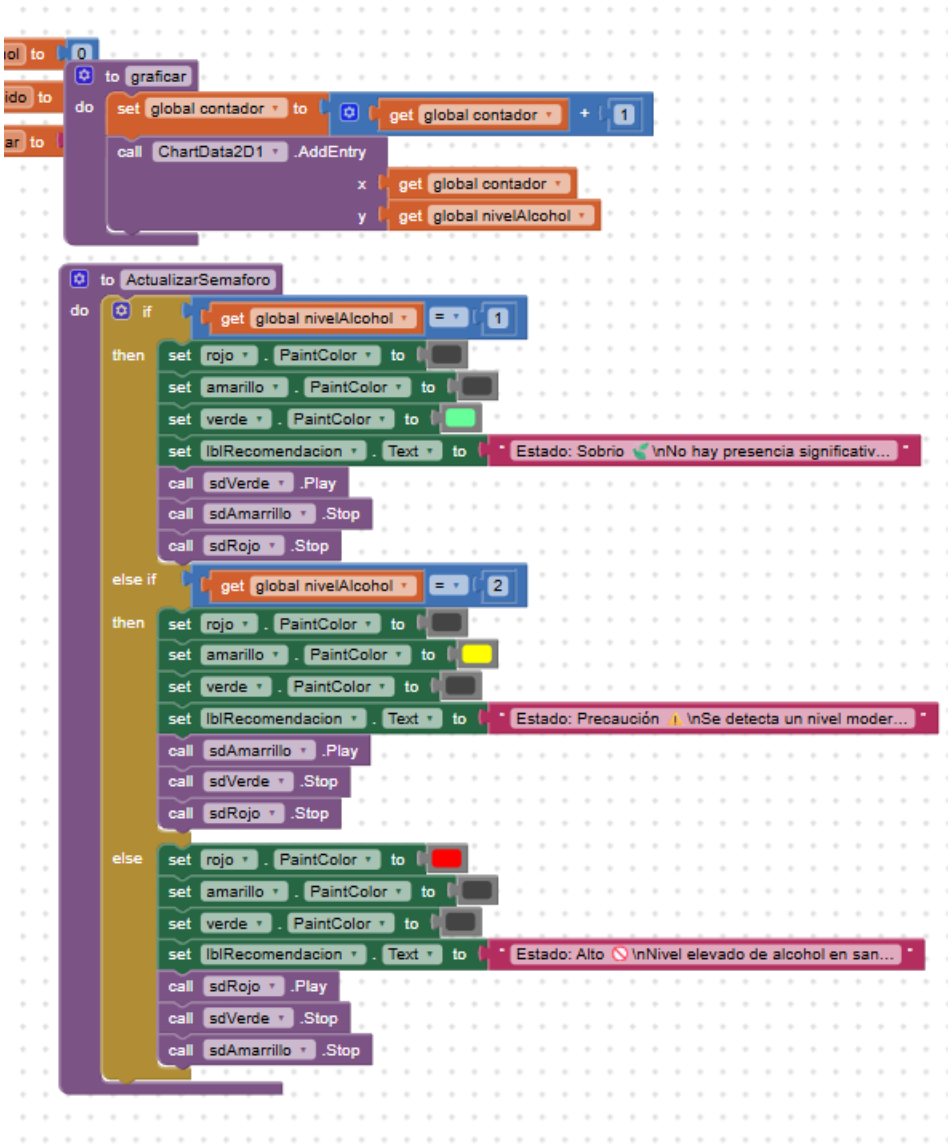
El conjunto de bloques de la imagen controla la conexión a Bluetooth, la carga del usuario y el cierre de sesión dentro de la pantalla del alcoholímetro. Cuando el usuario trate de conectarse sin elegir un método de conectividad, el sistema le indica un mensaje, un aviso para que busque un método de conectividad, y si intenta conectarse usando la dirección seleccionada, actualizando la etiqueta de estado según el resultado y mostrando mensajes de éxito o error. Los bloques BeforePicking y AfterPicking del ListPicker cargan la lista de dispositivos Bluetooth disponibles y guardan el seleccionado. Al iniciar la pantalla, la app carga el nombre del usuario que inició sesión y desactiva temporalmente el botón de detener. Finalmente, el botón **Cerrar Sesión** abre un cuadro de confirmación y, si el usuario elige "Sí", la app desconecta el Bluetooth y regresa a Screen1, cerrando así la sesión.

Luego están los bloques de los botones de iniciar, detener, guardar y ver historial, y el componente clock, que es fundamental para que la aplicación funcione, y los procesos de la gráfica y el semáforo.

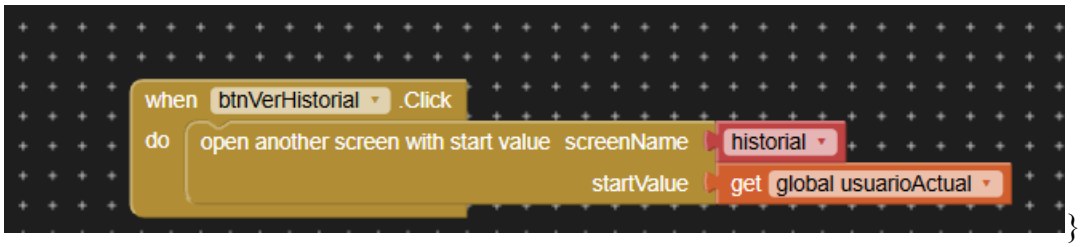


Este conjunto de bloques controla toda la lógica del alcoholímetro en App Inventor: al iniciar la prueba, la app verifica si el Bluetooth está conectado y envía el comando “I”, habilita el temporizador y comienza a recibir datos desde Arduino; el *Clock* procesa los valores recibidos, valida si son números, actualiza el nivel de alcohol, refresca el semáforo visual y grafica la medición; el botón Detener envía el comando “X”, desactiva el temporizador y vuelve a habilitar el botón de inicio; y el botón Guardar medición valida que exista un valor, envía el comando “S”, registra la fecha-hora, guarda el resultado en el historial del usuario y lo envía a la base de datos.

Bloques de semáforo y grafica

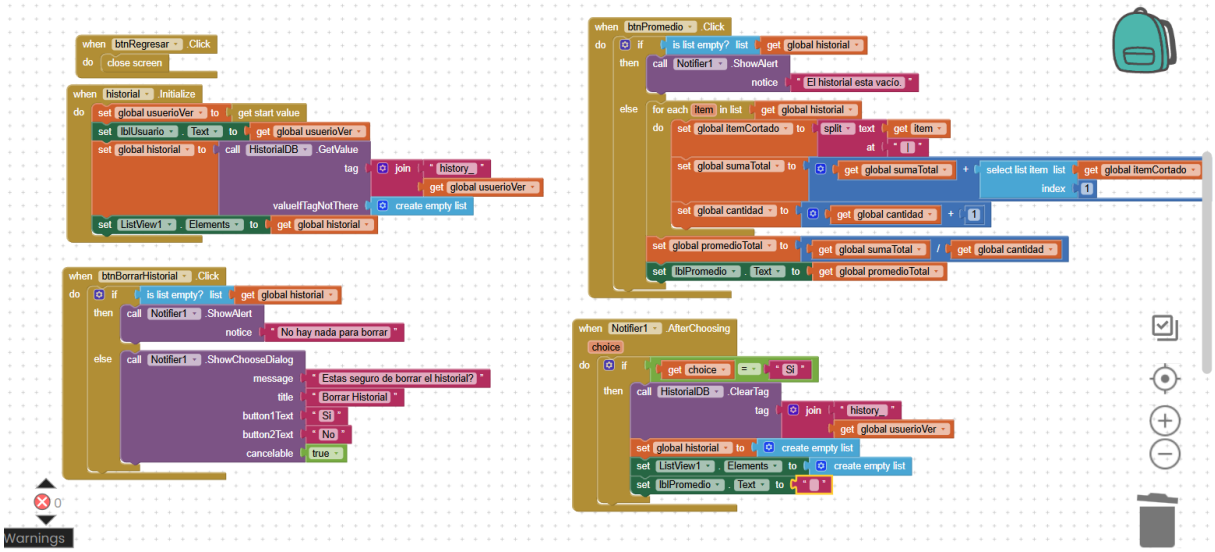


Según el nivel de alcohol en el aliento, el semáforo tomará su color respectivo, mostrará un mensaje de recomendación y sonará un audio con. Se graficará el nivel de alcohol para que el usuario vea sus estadísticas.



Al presionarlo, el usuario navegará a otra screen “Historial”, inicializando el nombre correspondiente al usuario.

4- Historial



Los bloques de la imagen gestionan la funcionalidad del usuario y el cálculo del promedio. Al inicializar la pantalla, se intenta cargar el historial correspondiente al usuario desde TinyDB bajo una etiqueta construida. El botón promedio calcula el promedio a partir del historial y el de borrar, borra los datos del historial.

4.1.2.5. Integración Hardware-Software

Protocolo de comunicación implementado:

El sistema utiliza un protocolo serial simple pero efectivo:

Formato de mensajes Arduino, App:

[VALOR_NUMERICO]\n

- VALOR_NUMERICO: Un dígito (1, 2 o 3)
- \n: Carácter de nueva línea (ASCII 10) como delimitador

Formato de mensajes App, Arduino:

[COMANDO_CHAR]

- COMANDO_CHAR: Un carácter ASCII ('I', 'X', 'S')

Comando	**Significado**	**Acción en Arduino**	**Respuesta esperada**
I o 1	Iniciar medición	pruebaActiva = true; recalibrar	Inicio de transmisiones periódicas
X o 2	Detener medición	pruebaActiva = false; apagar LEDs	Cese de transmisiones
S	Guardar registro	Ejecutar animación de confirmación	Flash de barra LED

Tabla 15: Comandos soportados

Gestión de errores y casos extremos:

Situación	Comportamiento del sistema
Conexión Bluetooth perdida durante medición	Arduino continúa operando localmente (LEDs siguen funcionando); App muestra alerta de desconexión
Dato corrompido recibido (ej: "2a3")	App valida con es_numero(), ignora el dato y espera siguiente transmisión válida
Múltiples comandos 'I' enviados consecutivamente	Arduino acepta cada comando y recalibra cada vez (comportamiento idempotente)
Buffer Bluetooth saturado	Delimitador \n garantiza que cada ReceiveText lee un mensaje completo, sin concatenación
Arduino reiniciado mientras App conectada	App detecta desconexión (IsConnected = false), usuario debe reconectar manualmente

Tabla 16: Gestión de incidencias

4.1.2.6 Diseño Físico del Producto Final

Especificaciones de la carcasa:

Aspecto	Especificación

Material	Cartón corrugado de doble pared (grosor 5mm)
Dimensiones externas	15.25 cm (L) × 20.15 cm (A) × 4.15 cm (H)
Peso total (con pilas)	~380 gramos
Color base	Gris oscuro con elementos gráficos en cian/amarillo/rojo
Acabado	Papel adhesivo fotográfico con diseño cyber-industrial

Tabla 17: Caja del proyecto/producto

Características estructurales:

1. Panel frontal:
    a. Apertura rectangular (8×2 cm) con lámina de acetato transparente para visualización de barra LED
    b. Logo B.A.R.T. en esquina superior derecha con tipografía técnica
2. Panel superior:

a. Rejilla circular (diámetro 2.5 cm) con malla metálica fina sobre el sensor MQ- 3

3. Panel lateral derecho:

a. Compartimento removible con tapa de velcro para pack de pilas AA

4. Panel posterior:

a. Orificio rectangular (1.5×0.8 cm) para acceso al puerto USB del Arduino

b. Etiqueta: "Puerto de programación - Solo personal técnico"

c. Switch ON/OFF (interruptor entre pack de pilas y jack barrel) para apagado completo

Organización interna:

Vista superior (tapa removida):

Fotografías del dispositivo final:

Figura 24: Imagen mockup principal de diseño

1. *Vista frontal con barra LED encendida*

Figura 25: Imagen de diseño de caja con diferentes niveles de leds activos

2. Vista superior mostrando sensor MQ-3

Figura 26: Imagen de la posición del sensor en caja.

3. Vista interna (sin tapa) mostrando organización de componentes

Figura 27: Imagen mockup del contenido electrónico interno de la caja.

4. Comparación de tamaño con objeto de referencia (ej: smartphone)

Figura 28: Imagen de dispositivo BART junto a un Nothing Phone 3 en comparación

4.1.2.7. Capturas de Pantalla de la Aplicación Móvil

Secuencia de uso completa:

1. *Inicio de sesión funcional*

Figura 29: Imagen de inicio de sesión en App

2. *Vista principal de la app*

Figura 30: Screenshot de la vista principal de la app funcional

3. ListPicker de dispositivos Bluetooth con IS-05 visible

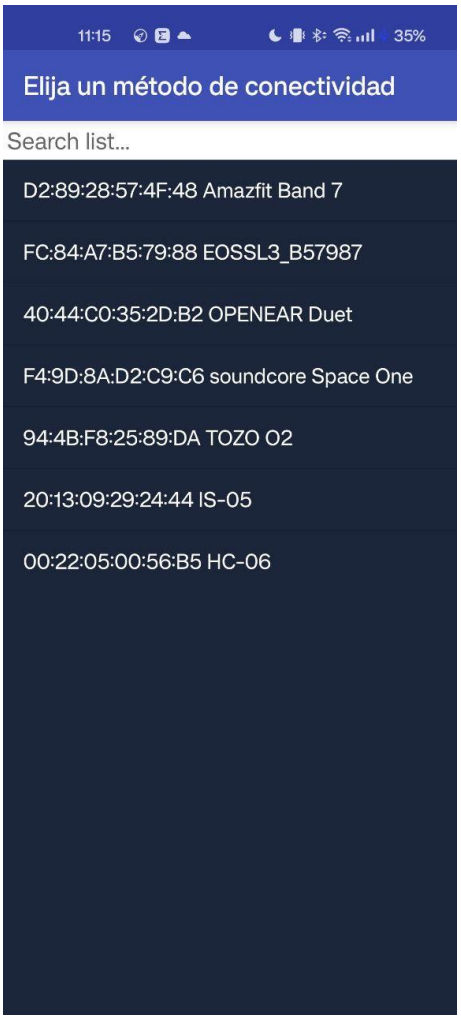


Figura 31: Menu de selección de conexión bluetooth

4.1.3 Evaluación

4.1.3.1. Evaluación de Desempeño Técnico

Prueba 1: Caracterización del sensor MQ-3

Se ejecutó el protocolo descrito en sección 3.3.3.1 con los siguientes resultados:

Métrica	Valor obtenido	Valor esperado	Estado
Línea base promedio	207 ADC	200-250 ADC	Dentro de rango
Desviación estándar (línea base)	4.2 ADC	<10 ADC	Baja variabilidad
Valor de pico promedio (5 mediciones)	682 ADC	>400 ADC	Respuesta fuerte
Coeficiente de variación	12.3%	<20%	Repetibilidad aceptable
Tiempo de onset (t_onset)	6.2 s	<15 s	Respuesta rápida
Tiempo de pico (t_peak)	11.8 s	<15 s	Dentro de especificación
Tiempo de recuperación (t_recovery)	74 s	<90 s	Recuperación adecuada

Interpretación: El sensor MQ-3 mostró comportamiento consistente con las especificaciones del datasheet. La repetibilidad con CV% de 12.3% es aceptable para un sensor de bajo costo destinado a pre-screening.

4.1.3.2. Evaluación de Usabilidad

Se ejecutaron las pruebas con 3 usuarios:

Perfil de participantes:

- Usuario 1: Estudiante de Ingeniería de Sistemas, 22 años, alta familiaridad con tecnología
- Usuario 2: Profesional de área administrativa, 35 años, uso básico de smartphone
- Usuario 3: Persona sin formación técnica, 52 años, uso ocasional de tecnología

Fortalezas identificadas:

1. Claridad de interpretación: Los 3 usuarios comprendieron correctamente el significado del semáforo (verde/amarillo/rojo) sin necesidad de explicación
2. Rapidez en tareas core: Las tareas críticas (iniciar medición, interpretar resultado) se completaron en <10 segundos promedio
3. Feedback visual efectivo: Los usuarios expresaron apreciación por las animaciones de LEDs y las notificaciones en la app

Debilidades identificadas (Pain Points): 1

Conexión Bluetooth (Tarea 2): Usuario 3 tuvo dificultad para encontrar el dispositivo "HC-05" en la lista de Bluetooth emparejados. Sugirió que el nombre debería ser "B.A.R.T." en lugar de "HC-05"

- Recomendación: Renombrar el módulo Bluetooth mediante comandos AT para facilitar identificación

2. Guardar medición (Tarea 6): Usuario 3 no identificó inmediatamente el botón "Guardar" entre los tres botones presentes

   a. Recomendación: Aumentar tamaño del botón o usar ícono de diskette universalmente reconocido

3. Falta de indicador de batería: Usuario 2 preguntó cómo saber cuándo cambiar las pilas

   a. Recomendación: Implementar en versión futura un divisor de voltaje para monitorear nivel de batería

Comentarios cualitativos destacados:

Usuario 1: ″Me sorprendió la velocidad de respuesta. Pensé que iba a ser más lento. Los LEDs le dan un toque profesional."

Usuario 2: ″Las recomendaciones en texto son muy claras. Esta bien que diga directamente si la persona puede trabajar o no, no tengo que adivinar.″

Usuario 3: ″Al principio me confundí con los botones de la app, pero después de usarlo una vez ya entendí. El semáforo es muy claro.″

Conclusión de evaluación:

B.A.R.T. ocupa exitosamente el nicho de herramienta educativa y de sensibilización para programas de SST con presupuesto limitado. No compite con dispositivos certificados,

sino que complementa el ecosistema de prevención al permitir evaluaciones frecuentes de bajo costo que fomentan cultura de autocontrol.

5. Conclusiones y trabajo futuro

5.1. Conclusiones

El proyecto BART ha logrado integrar un sistema de bajo costo para la monitorización de los niveles de alcohol en el aliento de las personas, cumpliendo con los objetivos definidos. El problema de adquisición, procesamiento y visualización remota de datos fue cubierto mediante la arquitectura de tres capas: el sensor MQ-3 para la adquisición, la plataforma Arduino para el filtrado, calibración y mapeo de estados lógicos (1, 2, 3), y la aplicación móvil (MIT App Inventor) como cliente de visualización y gestor de datos.

La solución desarrollada demostró ser funcional y robusta, verificando la viabilidad de la conexión Bluetooth Serial (HC-05) para la transferencia de datos en tiempo real.

El sistema ha logrado cumplir en:

1. Procesamiento de señal.

1.

Una arquitectura multi-usuario.

2.

Usabilidad.

3.

Control robusto.

5.2. Líneas de trabajo futuro

Las perspectivas de futuro del proyecto BART se centran en la expansión de su alcance y robustez, trascendiendo la limitación del entorno Bluetooth local. El desarrollo más inmediato consiste en la migración a una arquitectura de Internet de las Cosas y el

almacenamiento de datos históricos en servicios en la nube. Esto habilitaría funcionalidades de valor añadido, como la monitorización a distancia y la generación automatizada de reportes para el análisis longitudinal del historial de consumo; además, se considera la aplicación de técnicas de calibración inteligente para mitigar la dependencia del sensor MQ-3 a las variables ambientales.

AVISO LEGAL Y DE USO

El sistema B.A.R.T. (Biometric Alcohol Real-time Tracker) es un prototipo académico desarrollado con fines educativos y de investigación. Este dispositivo NO es un instrumento de medición certificado y NO tiene validez legal para procedimientos evidenciales o disciplinarios.

El dispositivo NO reemplaza equipos certificados por autoridades competentes (NHTSA, DOT, INVIMA) y su uso se limita a aplicaciones de sensibilización, prevención y autocontrol en programas de Seguridad y Salud en el Trabajo (SST).

Los autores no se responsabilizan por el uso indebido del sistema, decisiones basadas exclusivamente en sus mediciones, ni por daños derivados de su implementación fuera del contexto académico autorizado.

Para uso institucional o comercial, consulte con profesionales certificados y equipos que cumplan con la normativa vigente (Resolución 1843 de 2025 del Ministerio del Trabajo de Colombia).

© 2025 Pablo Trujillo Artunduaga y Brayan Toro Bustos.

Universidad Surcolombiana. Todos los derechos reservados.

6. Bibliografía

Colombia, Ministerio del Trabajo. (1950). Código Sustantivo del Trabajo [Decreto Ley 2663 de 1950]. Diario Oficial. https://www.mintrabajo.gov.co/normatividad/codigo-sustantivo-del-trabajo

Colombia, Ministerio del Trabajo. (2025). Resolución 1843 de 2025. Por la cual se establecen disposiciones relacionadas con los exámenes médicos ocupacionales. Ministerio del Trabajo de Colombia.

Colombia, Congreso de la República. (2012). Ley 1581 de 2012. Por la cual se dictan disposiciones generales para la protección de datos personales. Diario Oficial No. 48.587. http://www.secretariasenado.gov.co/senado/basedoc/ley_1581_2012.html

Comisión de Regulación de Comunicaciones - CRC. (2016). Resolución CRC 5050 de 2016. Por la cual se regula el uso de dispositivos inhibidores de señales de telecomunicaciones. Ministerio de Tecnologías de la Información y las Comunicaciones.

Fine, G. F., Cavanagh, L. M., Afonja, A., & Binions, R. (2010). Metal oxide semi-conductor gas sensors in environmental monitoring. Sensors, 10(6), 5469-5502. https://doi.org/10.3390/s100605469

Fondo de Riesgos Laborales. (2022). Estadísticas de accidentalidad laboral relacionada con consumo de sustancias psicoactivas en Colombia. Ministerio de Salud y Protección Social.

Fraden, J. (2016). Handbook of Modern Sensors: Physics, Designs, and Applications (5th ed.). Springer International Publishing. https://doi.org/10.1007/978-3-319-19303-8

Grand View Research. (2024). Wearable Technology Market Size, Share & Trends Analysis Report. Grand View Research, Inc. https://www.grandviewresearch.com/industry-analysis/wearable-technology-market

Hanwei Electronics. (2014). MQ-3 Alcohol Gas Sensor Technical Data (Version 1.3) [Datasheet]. https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-3%20ver1.3%20-%20Manual.pdf

Ministerio de Salud y Protección Social. (2023). Informe sobre consumo de sustancias psicoactivas y accidentalidad laboral en Colombia. Observatorio Nacional de Salud.

Ministerio de Tecnologías de la Información y las Comunicaciones - MinTIC. (2024). Boletín Trimestral de las TIC: Cifras del sector de telecomunicaciones. Gobierno de Colombia. https://www.mintic.gov.co/portal/inicio/

MIT App Inventor. (2023). MIT App Inventor Documentation and Tutorials. Massachusetts Institute of Technology. http://ai2.appinventor.mit.edu/reference/

Wolber, D., Abelson, H., Spert, E., & Looney, L. (2011). App Inventor: Create Your Own Android Apps. O'Reilly Media, Inc.

World Health Organization - WHO. (2018). Global Status Report on Alcohol and Health 2018. World Health Organization. https://www.who.int/publications/i/item/9789241565639

Arduino LLC. (2024). Arduino Reference: Language Reference. https://docs.arduino.cc/learn/built-in-libraries/software-serial/

Dräger Safety AG & Co. KGaA. (2023). Alcotest 7510 Technical Specifications. Dräger Safety.

Massachusetts Institute of Technology. (2024). MIT App Inventor Documentation. MIT Media Lab. http://ai2.appinventor.mit.edu/reference/

Naylamp Mechatronics. (2018). Tutorial sensores de gas MQ2, MQ3, MQ7 y MQ135. https://naylampmechatronics.com/blog/42_tutorial-sensores-de-gas-mq2-mq3-mq7-y-mq135.html

SparkFun Electronics. (2024). MQ-3 Alcohol Gas Sensor Hookup Guide. https://cdn.sparkfun.com/assets/6/a/1/7/b/MQ-3.pdf

Stoffregen, P. (2023). SoftwareSerial Library for Arduino [Software]. GitHub. https://github.com/PaulStoffregen/SoftwareSerial

BricoGeek. (2024). Alcoholímetro MQ-3: Ficha técnica del producto.

https://tienda.bricogeek.com/descatalogado/37-alcoholimetro-mq-3.html

Scribd. (2018). Alcohol Sensor: Technical Documentation [Documento técnico].

https://es.scribd.com/document/205987334/Alcohol-Sensor

Scribd. (2019). Alcoholímetro: Implementación con Arduino [Documento técnico].

https://es.scribd.com/document/396022378/alcoholimetro

Scribd. (2024). Alcoholímetro: Guía de construcción [Documento técnico]. https://es.scribd.com/document/714370246/Alcoholimetro-1

Gzz, R. (2012, mayo 15). Proyecto Arduino: Alcoholímetro. Ramon-Gzz Blog. https://ramon-gzz.blogspot.com/2012/05/proyecto-arduino-alcoholimetro.html

Instructables. (2020). Using MIT App Inventor to Control Arduino: The Basics. https://www.instructables.com/Using-MIT-App-Inventor-to-Control-Arduino-the-Basi/

Last Minute Engineers. (2023). MQ-3 Alcohol Sensor with Arduino – Complete Tutorial. https://lastminuteengineers.com/mq3-alcohol-sensor-arduino-tutorial/

RogerBit. (2017, enero 10). Alcoholímetro casero – Muy fácil. RogerBit Electronics. https://rogerbit.com/wprb/2017/01/alcoholimetro-casero/App Inventor y Arduino (Documento Google). (s.f.).

Arduino Forum. (2023). SoftwareSerial.h: No such file or directory [Hilo de discusión]. https://forum.arduino.cc/t/softwareserial-h-no-such-file-or-directory/1094733

Arduino Forum. (2022). Why I can't find the SoftwareSerial library and how to use the SomeSerial lib [Hilo de discusión]. https://forum.arduino.cc/t/why-i-cant-find-the-softwareserial-library-and-how-to-use-the-someserial-lib/675642

ArduWiki. (2024). SoftwareSerial Library Reference. https://arduwiki.perut.org/index.php/SoftwareSerial

App Inventor y Arduino – Documento adicional. (s.f.). https://docs.google.com/document/d/1OHrtagdggqEFZEzTndfo-TMvXDhbF5vlh3QD21r8WNw/edit

Gzz, R. (2012). *Proyecto Arduino: Alcoholímetro*. Blogspot. https://ramon-gzz.blogspot.com/2012/05/proyecto-arduino-alcoholimetro.html

Instructables. (n.d.). *Using MIT App Inventor to Control Arduino: The Basics*. https://www.instructables.com/Using-MIT-App-Inventor-to-Control-Arduino-the-Basi/

Last Minute Engineers. (n.d.). *MQ-3 Alcohol Sensor With Arduino – Complete Tutorial*. https://lastminuteengineers.com/mq3-alcohol-sensor-arduino-tutorial/

Lingoes Translator – Código de idiomas. (n.d.). *Lingoes*. http://www.lingoes.net/en/translator/langcode.htm

Moshiko. (2020, marzo 8). Construye un alcoholímetro con Arduino – Sensor MQ2 [Video]. YouTube. https://www.youtube.com/watch?v=9uY1H7X56t4

TechTutorials. (2023). Sensor MQ-3 alcoholímetro o detector de alcohol en Arduino UNO R3 [Video]. YouTube. https://www.youtube.com/watch?v=0DPsvl0d5o8

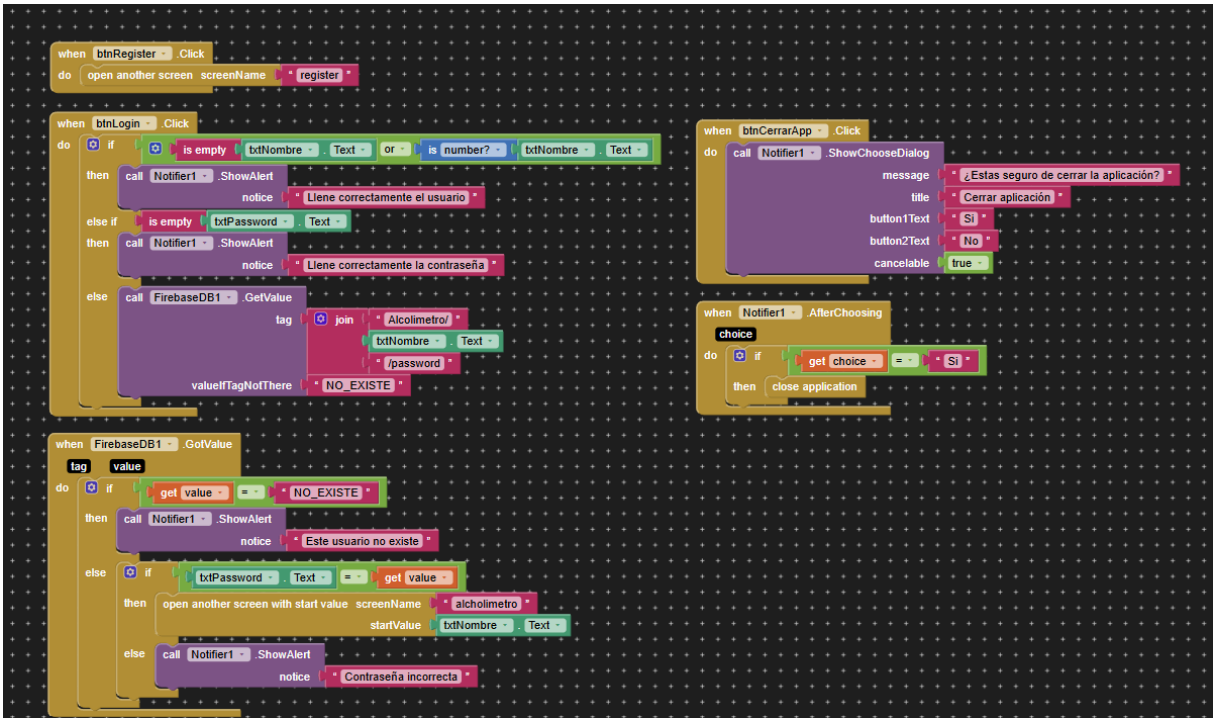
DIY Electronics. (2022). Alcoholímetro casero – Muy sencillo [Video]. YouTube. https://www.youtube.com/watch?v=2afC1Mc5Aec

Alcoholímetro – Tutorial adicional. (s.f.). *Scribd*. https://es.scribd.com/document/205987334/Alcohol-Sensor

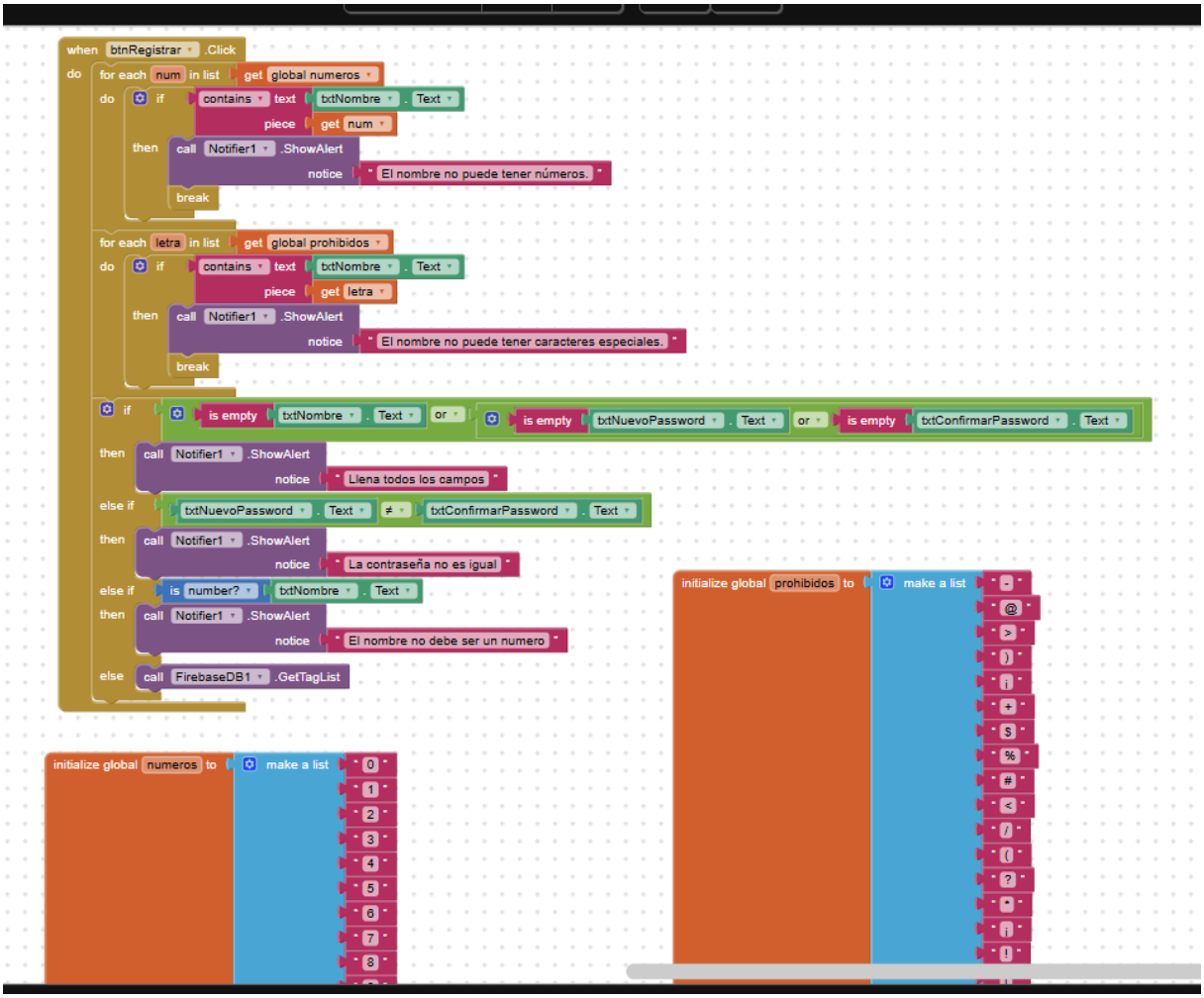
Toro Bustos, B., & Trujillo Artunduaga, P. (2025). Project BART: Biometric Alcohol Real-time Tracker [Repositorio de código]. GitHub. https://github.com/trbureiyan/Project-BART

Anexos

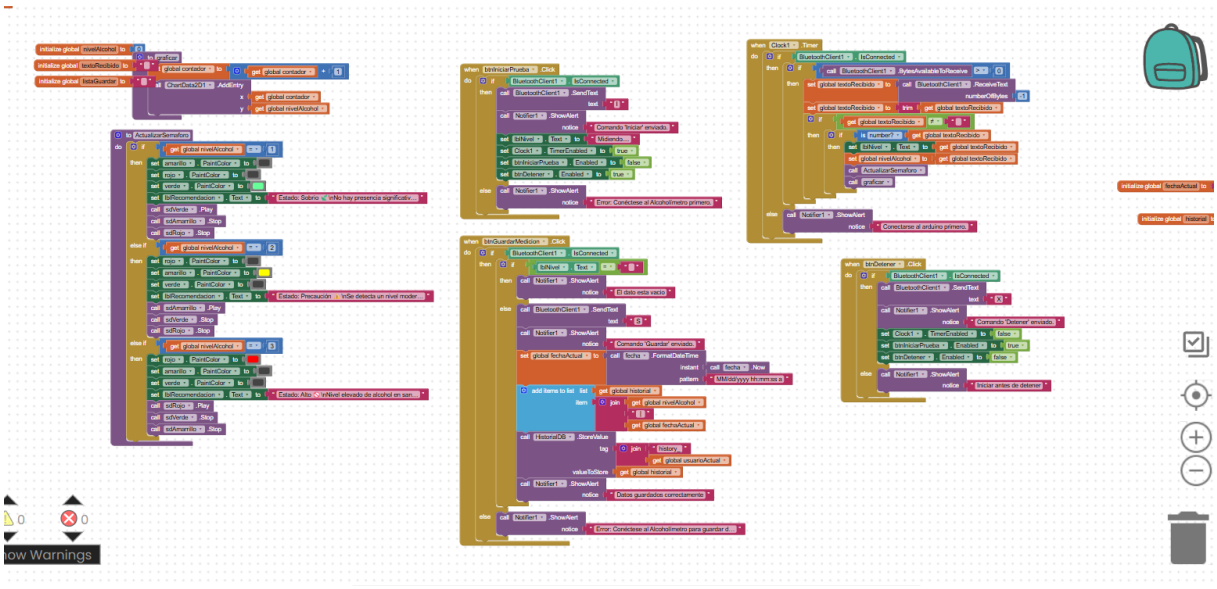
Anexo A. Bloques del login

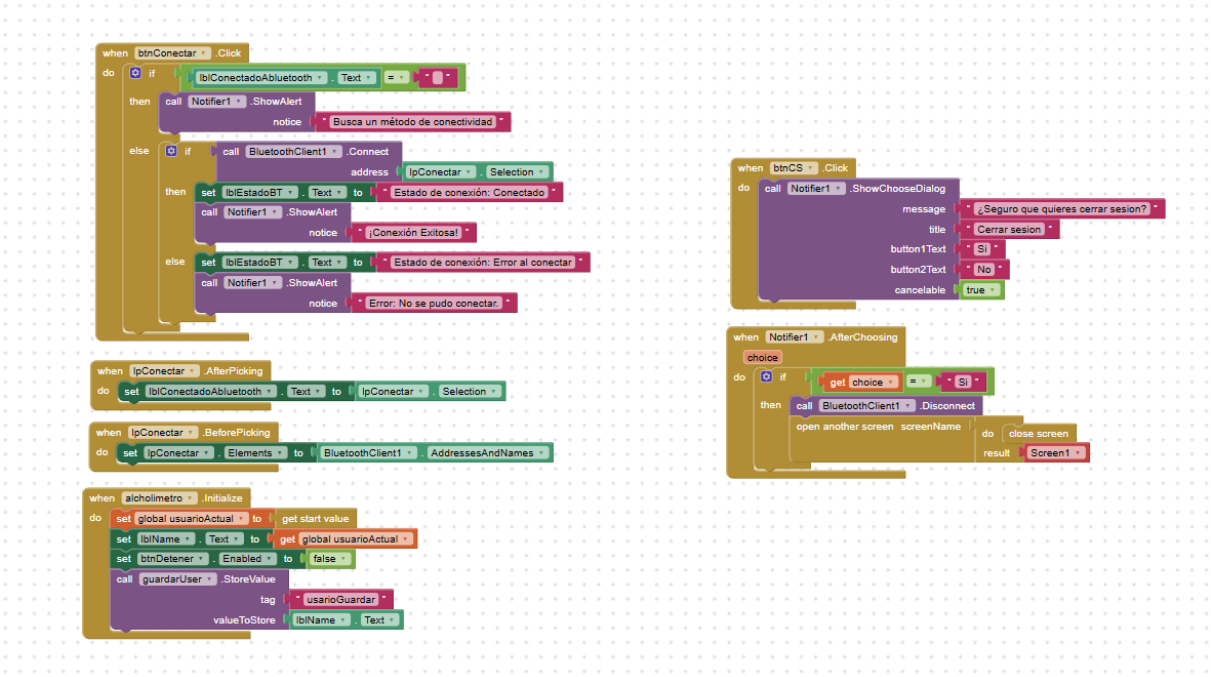


Anexo B. Bloques de register.

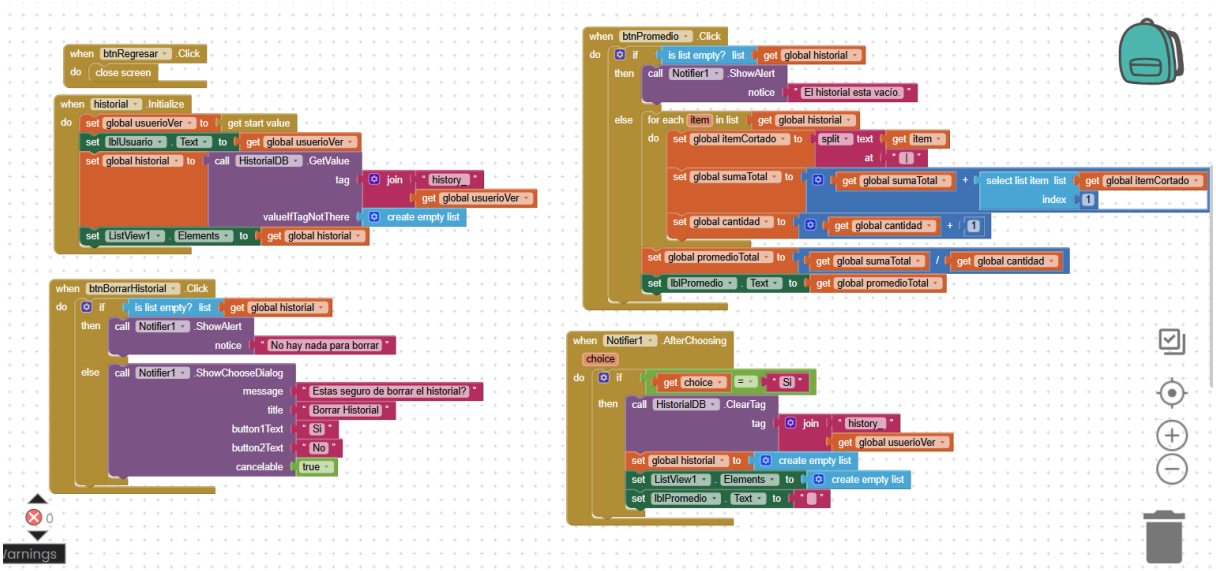


Anexo C. Bloques del alcoholímetro.





Anexo D. Bloques del historial.



Anexo E: Código Fuente del Firmware Arduino

Archivo: Alcoholimetro_BART_Ultimate.ino

Descripción: Código completo del firmware desarrollado para Arduino UNO Rev3, incluyendo módulos de calibración automática, filtrado de señal, gestión de comunicación Bluetooth y control de actuadores LED.

Lenguaje: C++ (Arduino Framework)

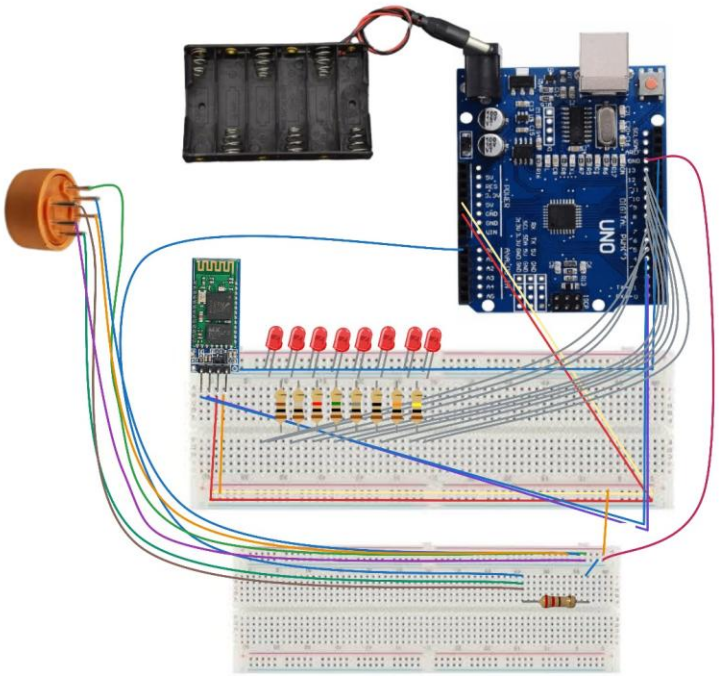
Líneas de código: 269 líneas

Repositorio: https://github.com/trbureiyan/Project-BART/blob/main/main.ino

Anexo F: Diagramas Técnicos

C.1. Diagrama Esquemático del Circuito

Archivo: Esquematico_BART_Circuito.png



Descripción: Representación técnica completa de las conexiones eléctricas del sistema, incluyendo valores de componentes pasivos, referencias de pines y especificaciones de voltaje.

Software utilizado: Excalidraw

Elementos representados:

Sensor MQ-3 (6 pines) con circuito de calentamiento y divisor de voltaje

Arduino UNO Rev3 con mapeo completo de pines

Módulo Bluetooth HC-05 (conexiones TX/RX alternativas)

Barra de 8 LEDs con resistencias limitadoras

Sistema de alimentación de 6 pilas AA con regulación a 5V

F.2. Diagrama de Arquitectura del Sistema

Archivo: Arquitectura_BART_3Capas.png

Descripción: Representación visual de la arquitectura de tres capas del sistema (Sensórica, Procesamiento, Presentación) con flujos de datos y protocolos de comunicación.

Elementos incluidos:

Capa 1: Sensórica analógica (MQ-3, acondicionamiento de señal)

Capa 2: Procesamiento digital (Arduino, firmware, algoritmos)

Capa 3: Interfaz de usuario (App móvil, visualización, almacenamiento)

Protocolos de comunicación: Serial UART, Bluetooth SPP

F.3. Diagrama de Flujo del Firmware

Archivo: Flujograma_Firmware_BART.png

Descripción: Representación detallada de la lógica de control del firmware Arduino, incluyendo procesos de inicialización, bucle principal y manejo de comandos.

Procesos documentados:

Rutina de setup(): Inicialización de puertos, calibración automática, animaciones de inicio

Rutina de loop(): Polling de comandos, adquisición de datos, procesamiento, transmisión

Funciones auxiliares: Filtrado de señal, mapeo de niveles, control de LEDs, gestión de comunicación

Anexo G: Fotografías del Prototipo

G.1. Vistas Externas del Dispositivo

Archivos:

BART_Vista_Frontal_LEDs.jpg: Panel frontal con barra LED encendida mostrando nivel 5

BART_Vista_Superior_Sensor.jpg: Panel superior con rejilla de ventilación sobre sensor MQ-3

BART_Vista_Lateral.jpg: Compartimento de pilas AA con tapa removible

BART_Vista_Posterior_Puertos.jpg: Panel posterior con acceso a puerto USB y switch de encendido

G.2. Vistas Internas del Dispositivo

Archivos:

BART_Interior_General.jpg: Vista completa del interior sin tapa, mostrando disposición de componentes

BART_Interior_Cableado.jpg: Detalle de organización de cables con código de colores y bridas

BART_Interior_Arduino.jpg: Montaje del Arduino UNO con protoboard y conexiones

BART_Interior_Sensor.jpg: Posicionamiento del sensor MQ-3 en zona de ventilación